Intégration CMOS analogique de réseaux de neurones à cliques

Benoit Larras

THÈSE / Télécom Bretagne
sous le sceau de l'Université européenne de Bretagne

pour obtenir le grade de Docteur de Télécom Bretagne
En accréditation conjointe avec l'Ecole Doctorale Sicma
Mention : Sciences et Technologies de l'Information et de la Communication

présentée par
Benoît Larras
préparée dans le département Electronique
Laboratoire Labsticc

Intégration CMOS analogique de réseaux de neurones à cliques

Thèse soutenue le 03 décembre 2015
Devant le jury composé de :

Patrick Loumeau
Professeur, Télécom ParisTech / président

Olivier Romain
Professeur, ENSEA – Lab. ETIS - Cergy / rapporteur

Sylvain Saïghi
Maître de conférences (HDR), Laboratoire IMS - Bordeaux / rapporteur

Simon Thorpe
Directeur de recherche, Université de Toulouse 3 / examinateur

Claude Berrou
Professeur, Télécom Bretagne / examinateur

Matthieu Arzel
Maître de conférences, Télécom Bretagne / examinateur

Fabrice Seguin
Maître de conférences, Télécom Bretagne / examinateur

Cyril Lahuec
Maître de conférences (HDR), Télécom Bretagne / directeur de thèse

N ° d'ordre : 2015telb0397

Sous le sceau de l'Université européenne de Bretagne

Télécom Bretagne

En accréditation conjointe avec l'École Doctorale Sicma

Intégration CMOS analogique de réseaux de neurones à cliques

Thèse de Doctorat

Mention : "STIC"

Présentée par **Benoît Larras**

Département : ELEC

Laboratoire : Lab-STICC Pôle : IAS

Directeur de thèse : **Cyril Lahuec**

Soutenue le 3 décembre 2015

Jury :

M. Olivier Romain,	Professeur des Universités, Université de Cergy-Pontoise	Rapporteur
M. Sylvain Saïghi,	Maître de Conférences HDR, Université de Bordeaux	Rapporteur
M. Patrick Loumeau,	Professeur, Télécom ParisTech	Examinateur
M. Simon Thorpe,	Directeur de recherche CNRS, Université de Toulouse III	Examinateur
M. Claude Berrou,	Professeur, Télécom Bretagne	Examinateur
M. Cyril Lahuec,	Maître de Conférences HDR, Télécom Bretagne	Directeur de thèse
M. Matthieu Arzel,	Maître de Conférences, Télécom Bretagne	Encadrant
M. Fabrice Seguin,	Maître de Conférences, Télécom Bretagne	Encadrant

Remerciements

Ce mémoire est le résultat de travaux de recherche de près de trois ans. En préambule, je veux adresser tous mes remerciements aux personnes avec lesquelles j'ai pu échanger et qui m'ont aidé dans la réalisation de ces travaux.

Tout d'abord, j'aimerais remercier l'ensemble des membres de mon jury de thèse : Messieurs Sylvain Saïghi, Olivier Romain, Simon Thorpe, Patrick Loumeau, Claude Berrou, Fabrice Seguin, Matthieu Arzel et Cyril Lahuec. J'ai réellement apprécié échanger avec vous pendant et après ma soutenance de thèse.

Je tiens également à remercier l'équipe NeuCod, et en particulier Claude Berrou, de m'avoir offert l'opportunité de travailler au sein de ce projet. De plus, je remercie tout le personnel du département ELEC de Télécom Bretagne (y compris les néo-retraités), pour votre accueil ainsi que pour toute la sympathie dont vous avez fait preuve à mon égard au cours de ces trois années. Ça a été un véritable plaisir de travailler dans un tel environnement, et je pense sincèrement qu'il m'a aidé à m'épanouir davantage.

Ensuite, merci à tous mes collègues thésards (encore en thèse ou diplômés) pour votre soutien et votre amitié. Merci tout particulièrement à Paul, André, Franck, Valentin, Tristan, Julie et Sébastien.

Il serait bien sûr ingrat de ne pas citer dans ces remerciements ma famille et mes amis. Vous m'avez toujours soutenu, dans les moments où tout allait bien comme dans les moments plus compliqués, et je vous en suis très reconnaissant.

Je tiens pour finir à remercier l'ensemble de mon équipe encadrante : mes encadrants tout d'abord, Fabrice Seguin et Matthieu Arzel, ainsi que mon directeur de thèse Cyril Lahuec. Merci beaucoup pour votre patience à mon égard, vos conseils tout comme votre support. Ça a été une grande joie et un plaisir non dissimulé de collaborer avec vous, et encore une fois je souhaite vous remercier pour tout ce que vous avez fait pour moi.

Ce travail a été financé par le Conseil Européen de la Recherche (bourse ERC-AdG2011 290901 NEUCOD).

Résumé

Les réseaux de neurones artificiels permettent de résoudre des problèmes que des processeurs classiques ne peuvent pas résoudre sans utiliser une quantité considérable de ressources matérielles. L'analyse et la classification de multiples signaux en sont des exemples. Ces réseaux sont de plus en plus implantés sur des circuits intégrés. Ils ont ainsi pour but d'augmenter les capacités de calcul de processeurs ou d'effectuer leur traitement dans des systèmes embarqués.

Dans un contexte d'application embarquée, la surface et la consommation d'énergie du circuit sont prépondérantes. Cependant, le nombre de connexions entre les neurones est élevé. De plus, les poids synaptiques ainsi que les fonctions d'activation utilisées rendent les implantations sur circuit complexes. Ces aspects, communs dans la plupart des modèles de réseaux de neurones, limitent l'intégration d'un réseau contenant un nombre de neurones de l'ordre de la centaine.

Le modèle des réseaux de neurones à cliques permet de réduire la densité de connexions au sein d'un réseau, tout en gardant une capacité de stockage d'information plus grande que les réseaux de Hopfield, qui est un modèle standard de réseaux de neurones. Ce modèle est donc approprié pour implanter un réseau de grande taille, à condition de l'intégrer de façon à garder la faible complexité de ses fonctions, pour consommer un minimum d'énergie.

Dans ce document, nous proposons un circuit mixte analogique/numérique implantant le modèle des réseaux de neurones à cliques. Nous proposons également plusieurs architectures de réseau pouvant contenir un nombre indéterminé de neurones. Cela nous permet de construire des réseaux de neurones à cliques contenant jusqu'à plusieurs milliers de neurones et consommant peu d'énergie.

Pour valider les concepts décrits dans ce document, nous avons fabriqué et testé un prototype d'un réseau de neurones à cliques contenant trente neurones sur puce. Nous utilisons pour cela la technologie Si CMOS 65 nm, avec une tension d'alimentation de 1 V. Le circuit a des performances de récupération de l'information similaires à celles du modèle théorique, et effectue la récupération d'un message en 58 ns. Le réseau de neurones occupe une surface de silicium de 16 470 μm^2 et consomme 145 μW. Ces mesures attestent une consommation d'énergie par neurone de 423 fJ au maximum. Ces résultats montrent que le circuit produit est dix fois plus efficace qu'un équivalent numérique en termes de surface de silicium occupée et de latence.

Mots-clés : Réseaux de neurones artificiels, réseaux de neurones à cliques, implantation mixte analogique/numérique.

Abstract

Artificial neural networks solve problems that classical processors cannot solve without using a huge amount of resources. For instance, multiple-signal analysis and classification are such problems. Moreover, artificial neural networks are more and more integrated on-chip. They aim therefore at increasing processors computational abilities or processing data in embedded systems.

In embedded systems, circuit area and energy consumption are critical parameters. However, the amount of connections between neurons is very high. Besides, circuit integration is difficult due to weighted connections and complex activation functions. These limitations exist for most artificial neural networks models and are thus an issue for the integration of a neural network composed of a high number of neurons (hundreds of them or more).

Clique-based neural networks are a model of artificial neural networks reducing the network density, in terms of connections between neurons. Its information storage capacity is moreover greater than that of a standard artificial neural networks model such as Hopfield neural networks. This model is therefore suited to implement a high number of neurons on chip, leading to low-complexity and low-energy consumption circuits.

In this document, we introduce a mixed-signal circuit implementing clique-based neural networks. We also show several generic network architectures implementing a network of any number of neurons. We can therefore implement clique-based neural networks of up to thousands of neurons consuming little energy.

In order to validate the proposed implementation, we have fabricated a 30-neuron clique-based neural network prototype integrated on chip for the Si 65-nm CMOS 1-V supply process. The circuit shows decoding performances similar to the theoretical model and executes the message recovery process in 58 ns. Moreover, the entire network occupies a silicon area of 16,470 μm^2 and consumes 145 μW, yielding a measured energy consumption per neuron of 423 fJ maximum. These results show that the fabricated circuit is ten times more efficient in terms of occupied silicon area and latency than a digital equivalent circuit.

Keywords : Artificial neural networks, clique-based neural networks, mixed analog/digital circuit implementation.

Table des matières

Table des figures

Liste des tableaux

Notations

Dimensionnement physique d'un réseau à cliques

$A_{fanaux_{tot}}$	Aire occupée par les fanaux dans le réseau
A_{h_conn}	Aire occupée par les connexions horizontales dans un cluster
A_{mem_conn}	Aire qu'occupe la mémoire des connexions si l'on doit les stocker dans le réseau
A_{mem_etats}	Aire qu'occupe la mémoire des états des fanaux si l'on doit les stocker dans le réseau
A_{reseau}	Aire du réseau
$A_{routage}$	Aire de l'unité de routage permettant la décomposition des itérations du processus de récupération de messages
$A_{synapses_{tot}}$	Aire occupée par les synapses dans le réseau
A_{v_conn}	Aire occupée par les connexions verticales dans un cluster
L_{FA}	Longueur de la partie analogique d'un fanal
L_{FD}	Longueur de la partie numérique d'un fanal
L_H	Espacement entre deux connexions horizontales
L_{SA}	Longueur de la partie analogique d'une synapse
L_V	Espacement entre deux connexions verticales
W_A	Largeur de la partie analogique d'un fanal ou d'une synapse
W_D	Largeur de la partie numérique d'un fanal

Paramètres d'un réseau de neurones à cliques

d	Densité du réseau à cliques
$E_{i,j}$	Valeur de l'entrée $\#j$ du fanal $\#i$
E_{ext_i}	Valeur de l'entrée de stimulation externe du fanal $\#i$
N_{bits}	Nombre de bits d'information portés par chaque fanal
N_{bits_max}	Nombre de bits d'information maximum pouvant être stockés dans le réseau
N_C	Nombre de clusters dans un réseau à cliques
N_{conn_eff}	Nombre de connexions effectivement réalisées dans le réseau
N_{conn_tot}	Nombre total de connexions pouvant être réalisées dans le réseau
N_F	Nombre de fanaux par cluster
$N_{F_{tot}}$	Nombre de fanaux dans un réseau à cliques
N_{IT}	Nombre d'itérations du processus de récupération de messages
$N_{messages}$	Nombre de messages stockés dans le réseau
N_P	Nombre de clusters implantés en parallèle dans une organisation basée sur la réutilisation matérielle
N_S	Nombre de synapses par fanal
$N_{S_{tot}}$	Nombre de synapses dans un réseau à cliques
S_i	Valeur de la sortie du fanal $\#i$
$T_{addition}$	Temps de réponse d'un additionneur
$T_{cluster}$	Temps de réponse d'un cluster
T_{CONV}	Temps de convergence du réseau à cliques
T_h	Durée d'un cycle d'horloge pour un système synchrone
ν_i	Nombre de contributions actives dans le fanal $\#i$

Circuit électrique

C_{syn}	Capacité équivalente d'une synapse
C_{syn_on}	Valeur de la capacité équivalente d'une synapse active
C_{syn_off}	Valeur de la capacité équivalente d'une synapse inactive
I_{BIAS}	Courant de polarisation de la cellule de compensation des variations PVT
I_{MANUEL}	Courant de polarisation manuelle du seuil du buffer de décision
$I_{OUT_{WTA}}$	Courant circulant dans un élément gagnant de circuit WsTA dans [RH14]
I_{UNIT}	Courant unitaire pouvant circuler dans une synapse
I_{WTA}	Courant circulant dans un élément gagnant de circuit WTA dans [LRMM88]
L_{min}	Longueur minimale du canal d'un transistor
R_{syn}	Résistance équivalente d'une synapse
R_{syn_on}	Valeur de la résistance équivalente d'une synapse active
R_{syn_off}	Valeur de la résistance équivalente d'une synapse inactive
V_{CC}	Tension d'alimentation du circuit
$V_{commande}$	Tension de commande du buffer à seuil variable
V_{COMP}	Tension de comparaison d'un élément de circuit WTA
$V_{DS_{sat}}$	Tension drain-source de saturation d'un transistor MOS
V_{ref}	Tension de seuil référence du buffer à seuil variable

Signaux d'entrées/sorties du circuit intégré

$AVCC$	Alimentation du cœur analogique du circuit
$AVCCE$	Alimentation de l'anneau analogique de la puce
Bus_Commun_R3	Signaux des états des fanaux, communs aux clusters dans le réseau $R3$
$Clock$	Signal d'horloge des circuits synchrones de la puce
$Current\_Mode$	Signal de sélection du mode de fonctionnement du buffer de décision
$DVCC$	Alimentation du cœur numérique du circuit
$DVCCE$	Alimentation de l'anneau numérique de la puce
L_{IN}	Signal de début de récupération d'un message
L_{OUT}	Signal de mémorisation du résultat de la récupération d'un message
$Load_{IN}$	Signal de début d'une itération dans le réseau $R3$
$Load_{OUT}$	Signal de mémorisation des sorties des fanaux après chaque itération dans le réseau $R3$
LR_{IN}	Signal d'activation des registres SIPO
LR_{OUT}	Signal d'activation des registres PISO
MAJ	Signaux de sélection d'un cluster dans le réseau $R3$
Out_F1_R1	Signal de sortie du fanal #1 du réseau $R1$
Out_F2_R1	Signal de sortie du fanal #2 du réseau $R1$
$Reset$	Signal de remise à zéro du circuit
$Serial_{IN}$	Signal d'entrée (en série) du message à décoder
$Serial_{OUT}\_R2$	Signal de sortie (en série) du message décodé par le réseau $R2$
$Serial_{OUT}\_R3$	Signal de sortie (en série) du message décodé par le réseau $R3$
$Stim\_F1\_1\_R1$	Signal de stimulation #1 du fanal #1 du réseau $R1$
$Stim\_F1\_2\_R1$	Signal de stimulation #2 du fanal #1 du réseau $R1$
$Stim\_F2\_R1$	Signal de stimulation du fanal #2 du réseau $R1$

Liste des acronymes

ASIC	Application-Specific Integrated Circuit
CMOS	Complementary MOS
FPGA	Field Programmable Gate Array
HH	Hodgkin-Huxley
IF	Integrate-and-Fire
LIF	Leaky-Integrate-and-Fire
LsKO	Losers-Kicked-Out
MOS	Metal Oxide Semiconductor
MPSoC	Multiprocessor System-on-Chip
NMOS	Transistor MOS canal N
NoC	Network-on-Chip
PISO	Parallel In - Serial Out
PMOS	Transistor MOS canal P
PVT	Process-Voltage-Temperature
SIPO	Serial In - Parallel Out
SoC	System-on-Chip
WsTA	Winners-Take-All
WTA	Winner-Takes-All

Publications associées

Conférences

B. Larras, C. Lahuec, M. Arzel and F. Seguin. Analog implementation of encoded neural networks. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 1612-1615, May 2013.

B. Larras, B. Boguslawski, C. Lahuec, M. Arzel, F. Seguin and F. Heitzmann. Analog encoded neural network for power management in MPSoC. In *New Circuits and Systems Conference (NEWCAS), 2013 IEEE 11th International*, pages 1-4, June 2013.

Prix du deuxième meilleur papier étudiant à la conférence NEWCAS 2013.

B. Larras, C. Lahuec, M. Arzel and F. Seguin. Design of analog subthreshold encoded neural network circuit in sub-100nm CMOS. In *Neural Networks, 2015 (IJCNN 2015), International Joint Conference on*, July 2015.

Publications en revue

B. Larras, B. Boguslawski, C. Lahuec, M. Arzel, F. Seguin and F. Heitzmann. Analog encoded neural network for power management in MPSoC. *Analog Integrated Circuits and Signal Processing*, 81(3) :585-605, 2014.

B. Boguslawski, F. Heitzmann, **B. Larras** and F. Seguin. Energy efficient associative memory based on neural cliques. *Circuits and Systems II (TCAS II), IEEE Transactions on*, in press.

Introduction

Le traitement de l'information dans des domaines variés repose sur des unités de calcul et des processeurs de plus en plus performants. Cependant, malgré l'augmentation de leurs ressources de mémoire et calcul, ainsi que de leur vitesse d'exécution, certains problèmes ne peuvent toujours pas être résolus de manière simple. Par exemple, reconnaître un objet sur une image quelle que soit sa position ou son inclinaison, reconnaître une voix dans un environnement sonore et identifier les mots prononcés, ou encore acquérir et analyser de multiples signaux (comme les signaux électroencéphalogrammes, ou signaux EEG), sont autant de problèmes que les processeurs conventionnels ne peuvent résoudre qu'au prix de coûteux calculs.

Il convient alors de se demander si ce n'est pas le matériel disponible, mais la manière de traiter ces problèmes, qui est erronée. En effet, des approches classiques en traitement de l'information, comme l'approche algorithmique (produire un processus systématique permettant de résoudre un problème) ont des exécutions binaires. De tels programmes ne permettent donc pas de s'adapter à la survenue de situations imprévues par les concepteurs. Or, ces problèmes sont résolus spontanément par le cerveau des êtres vivants, notamment celui de l'être humain. Une approche alternative de traitement de l'information consiste donc à reproduire le comportement naturel du cerveau pour résoudre ces problèmes.

De nombreux modèles essayent de reproduire ce comportement : ce sont les réseaux de neurones artificiels. Depuis les premiers modèles de perceptrons simple couche développés en 1957 [Ros57] jusqu'aux réseaux de neurones convolutionnels mis en œuvre aujourd'hui [LBH15], différents types de réseaux de neurones ont été développés. Chaque modèle se base sur le comportement de différentes régions du cerveau, par exemple le système visuel ou encore la mémoire à long terme. En conséquence, les différents modèles de réseaux de neurones remplissent des rôles différents, comme ceux de classification de données ou de mémoire associative (mémoire adressable par contenu).

D'abord implémentés par des programmes informatiques, les réseaux de neurones artificiels sont de plus en plus intégrés physiquement pour augmenter les capacités de calcul de processeurs. C'est par exemple le cas du projet *SpinNaker* de l'université de Manchester [PPG$^+$13] ou encore du projet *TrueNorth* d'IBM [Mer14]. Des systèmes embarqués résolvant des problèmes complexes comme de la reconnaissance de formes ou de signaux ont aussi besoin de circuits intégrés spécialisés (ou ASIC pour *Application-Specific Integrated Circuit*), car ceux-ci consomment moins d'énergie que des processeurs, et ont des temps de traitement plus faibles.

Dans des applications comme la reconnaissance et la classification de signaux EEG par exemple, les signaux sont acquis par un nombre de capteurs pouvant aller jusqu'à plusieurs centaines. Les réseaux de neurones traitant ces signaux doivent donc contenir un nombre de neurones au moins du même ordre de grandeur. Cependant, la connectivité élevée entre les neurones dans un réseau et la consommation croissante d'énergie avec le nombre de neurones rendent l'intégration d'un réseau de cette taille complexe. Des circuits intégrant des réseaux de neurones de grande taille ayant une faible consommation de ressources matérielles et d'énergie (typiquement de l'ordre du milliwatt) sont donc nécessaires.

Certains modèles de réseaux de neurones s'affranchissant de la complexité physiologique de ce dernier permettent de simplifier la structure du circuit ainsi que les échanges d'information entre les neurones artificiels. De plus, l'utilisation de certaines techniques de conception d'électronique analogique, comme l'utilisation des transistors sous le seuil, permet de réduire la consommation énergétique d'un circuit. En combinant ces différentes méthodes, il est donc possible de concevoir des circuits à faible consommation d'énergie et de ressources matérielles.

Dans ce document, notre but est donc de proposer un nouveau circuit capable d'implanter un réseau de neurones comprenant des centaines, voire des milliers de neurones pour un budget de puissance de l'ordre du microwatt par neurone. Les enjeux de notre travail sont donc les suivants. Tout d'abord, l'implantation d'un grand nombre d'éléments pose des problèmes de consommation, tant en termes de ressources matérielles qu'en termes d'énergie. Nous devrons donc faire des choix afin de minimiser ces problèmes. Ces choix induisent la sélection d'un modèle de réseau de neurones composé de fonctions simples et à la connectivité entre neurones limitée, ainsi que la définition de l'organisation topologique des neurones entre eux. Le choix de la méthode d'intégration permet aussi de simplifier la représentation de l'information dans le réseau, tandis que des choix de conception de circuit simplifient l'intégration des fonctions mises en œuvre.

Le second enjeu de ce travail est de proposer un circuit flexible permettant de s'adapter à plusieurs applications. Cette flexibilité a deux aspects. Le premier concerne le fait de pouvoir traiter n'importe quel jeu de données avec le même réseau. Cela nécessite de réinitialiser le réseau pour pouvoir apprendre de nouvelles informations. Le second aspect de flexibilité est de pouvoir faire évoluer la structure du réseau, afin de s'appliquer à un autre contexte. L'enjeu est de passer à l'échelle de réseaux de tailles encore plus importantes en limitant les efforts de conception.

Enfin, le troisième enjeu de ce travail est de proposer un circuit robuste aux éléments parasites inhérents à un fonctionnement sur puce. Il arrive que, dans un circuit intégré de surface importante, les éléments constitutifs ne fonctionnent pas dans les mêmes conditions environnementales (comprenant la température et la tension d'alimentation). Comme nous cherchons à intégrer un grand nombre d'éléments, nous devons vérifier que ces conditions n'altèrent pas le fonctionnement du circuit.

Finalement, nous démontrons la fonctionnalité du circuit proposé et des concepts évoqués sur un prototype de réseau intégré sur puce, contenant plusieurs dizaines de neurones. Cette puce sert ainsi de preuve de concept pour notre travail. Le document est structuré de la manière suivante.

Dans le **Chapitre 1**, nous étudions les différents modèles de réseaux de neurones. Cela nous permet de déterminer lequel est le plus adapté à l'implantation d'un grand nombre de neurones. Nous devons aussi choisir une méthode d'intégration qui simplifie le fonctionnement d'un neurone, en nous éloignant du comportement physiologique de ce dernier. Une fois nos choix arrêtés, nous décrivons en détail l'organisation des neurones dans le modèle choisi, ainsi que les fonctions réalisées au sein du neurone.

En connaissant les éléments composant le réseau et la façon dont ils sont ordonnés, nous proposons dans le **Chapitre 2** une organisation topologique de ces éléments sur un circuit. Nous pouvons alors déterminer la complexité de l'intégration d'un réseau en fonction du nombre de neurones à intégrer. Enfin, nous cherchons à optimiser cette organisation en appliquant les principes de la réutilisation matérielle utilisés en électronique numérique.

Dans le **Chapitre 3**, nous nous intéressons aux fonctions élémentaires qui composent le cœur des neurones. Nous implantons ainsi chacune des fonctions et vérifions leur comportement en simulation. Nous les assemblons ensuite pour former un neurone complet. Nous vérifions enfin comment se comportent plusieurs neurones mis en concurrence.

Le **Chapitre 4** nous permet d'évaluer le comportement des neurones précédemment conçus dans un réseau complet. Nous définissons d'abord les métriques qui nous permettent d'évaluer les performances du réseau. Puis, nous dimensionnons un réseau destiné à être intégré sur puce, dont nous simulons les performances théoriques. Nous nous intéressons ensuite aux imperfections inhérentes à la réalisation d'un circuit intégré, comme les variations des paramètres environnementaux et le désappariement des transistors. Nous proposons alors des moyens de compenser les effets de ces imperfections sur le fonctionnement du circuit.

Nous présentons le circuit intégré ainsi que les mesures effectuées sur le prototype dans le **Chapitre 5**. Les résultats de ces mesures sont alors comparés aux résultats obtenus en simulation dans le Chapitre 4. Cela nous permet de valider le bon fonctionnement des différents éléments du circuit. Une comparaison est ensuite réalisée par rapport à un circuit numérique équivalent, en termes de latence et de surface de silicium occupée.

En conclusion, nous proposons des perspectives de travail pour étendre les résultats obtenus à un réseau plus grand. En se basant sur notre travail, il est ainsi possible de concevoir des réseaux de plusieurs milliers de neurones.

Chapitre 1

Réseaux de neurones artificiels

Introduction

En introduction de ce document, nous avons montré l'intérêt de reproduire le comportement du traitement de l'information par le cerveau humain. Cela peut être utilisé dans des applications qui nécessitent de nombreux traitements en parallèle, comme dans la classification d'images, par exemple. L'objectif est donc de trouver une méthode pour intégrer un réseau d'un grand nombre de neurones, c'est-à-dire des centaines voire des milliers. Dans ce chapitre, nous faisons, dans un premier temps, une description générale des réseaux de neurones artificiels. Nous faisons ensuite l'inventaire des modèles de réseaux de neurones artificiels. Puis, nous exposons les différentes méthodes d'intégration de ces modèles sur circuit, ainsi qu'un état de l'art des circuits intégrant des réseaux de neurones artificiels. L'étude de ces circuits nous permet de faire un choix sur le modèle que met en œuvre le circuit dans la suite de ce document. Dans un second temps, nous décrivons plus en détail le modèle que nous allons utiliser : les réseaux de neurones à cliques proposés dans [GB11]. Nous concluons enfin sur la méthode d'intégration la plus efficace pour intégrer sur puce ce type de réseau de neurones artificiels.

1.1 Concepts fondamentaux des réseaux de neurones artificiels

Dans cette section, nous décrivons plus en détail la composition d'un réseau de neurones artificiels. Ce dernier peut être vu comme un graphe contenant des nœuds, les neurones, et des arêtes, les connexions entre ces neurones. Chaque neurone se sert des informations qu'il reçoit provenant d'autres neurones pour générer sa propre information grâce à une *fonction d'activation*, et la propager dans le réseau. Les liens à travers lesquels ces informations transitent arrivent vers des *synapses* à l'entrée de chaque neurone. Il y a une synapse par connexion incidente dans un neurone. La passage dans les synapses pondère les informations par des *poids synaptiques*. Les valeurs de ces poids influent sur le résultat que fournit le réseau, et trouver les poids qui permettent d'obtenir le fonctionnement désiré s'appelle l'*apprentissage* du réseau. Les poids synaptiques peuvent être mis à jour pendant le fonctionnement en fonction des données traitées par le réseau.

Un réseau de neurones artificiel peut ainsi être caractérisé complètement grâce à trois paramètres :

- l'organisation du réseau, c'est-à-dire les motifs d'interconnexions entre les neurones ;
- la composition interne du neurone, et plus précisément sa fonction d'activation ;
- le type d'apprentissage utilisé pour mettre à jour les poids synaptiques.

Dans la suite de cette section, nous décrivons plus en détail chacun de ces paramètres.

1.1.1 Organisation d'un réseau de neurones artificiels

Un paramètre permettant de caractériser un réseau de neurones artificiels est l'organisation de ses neurones, et comment ces derniers sont connectés entre eux. L'application dans laquelle va être utilisé le réseau définit les particularités de l'organisation du réseau de neurones, c'est-à-dire son nombre de neurones, le type de connexions entre les neurones, etc.

Il existe deux façons d'organiser les neurones dans un réseau de neurones artificiels. Tout d'abord, les neurones peuvent être groupés en *couches* qui vont se transmettre leurs informations les unes aux autres. Les informations ne vont alors transiter que dans un sens. Une couche va recevoir des informations de la précédente, les traiter et les envoyer à la suivante. Les couches aux extrémités du réseau sont alors nommées couches *d'entrée* et *de sortie*, et forment les interfaces avec l'extérieur du réseau. Dans ce cas, il n'y a pas de connexion entre deux neurones d'une même couche, mais chaque neurone envoie ses informations à tous les neurones de la couche suivante. Les connexions sont alors unidirectionnelles. C'est le principe du réseau appelé *perceptron* décrit dans [Ros57], qui sert à classifier de manière binaire des signaux d'entrée dans plusieurs catégories ("est-ce que ce signal appartient à cette catégorie ou non ?").

La deuxième manière d'organiser un réseau de neurones est de connecter de façon complète les neurones. Cela signifie que chaque neurone est connecté à tous les autres dans le réseau. Dans ce cas, certains neurones appelés *neurones d'entrée* vont recevoir des données de l'extérieur. Puis, les informations vont transiter dans le réseau jusqu'à ce que ce dernier converge vers un état stable. Le résultat est alors fourni par des *neurones de sortie*. Dans ce type d'organisation, les connexions sont bidirectionnelles. Cette organisation est utilisée dans les réseaux appelés réseaux

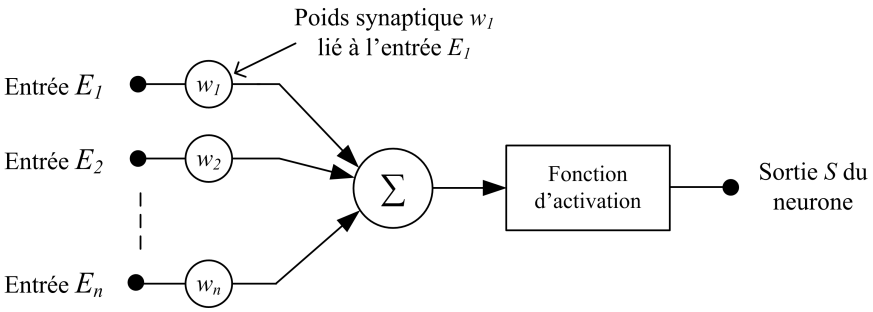

FIGURE 1.1 – Schéma de la structure d'un neurone artificiel. Les entrées sont pondérées par des poids synaptiques, puis additionnées. Le résultat de cette addition est ensuite appliqué à une fonction d'activation, dont le résultat décide de l'information envoyée par le neurone à sa sortie.

de Hopfield [Hop82], qui sont utilisés comme des mémoires associatives.

1.1.2 Composition d'un neurone artificiel

Le second paramètre permettant de caractériser un réseau de neurones artificiels est le type de fonction composant chaque neurone. Typiquement, un neurone artificiel est organisé comme dans la Figure 1.1.

Les informations d'entrée sont tout d'abord multipliées par les poids synaptiques associés à chaque connexion. Une fois pondérés, ces signaux sont additionnés et évalués à travers une fonction d'activation. Cette dernière peut varier d'un réseau de neurones artificiels à l'autre, selon le comportement que l'on veut donner aux neurones artificiels. La complexité de la fonction d'activation peut alors rapidement augmenter. En effet, si l'activation ne peut être conditionnée que par une simple comparaison par rapport à un seuil, elle peut aussi utiliser des mécanismes plus complexes comme la fonction *sigmoïde*, qui fait intervenir une fonction tangente hyperbolique dans son expression. Le résultat est ensuite propagé dans le reste du réseau.

1.1.3 Notion d'apprentissage

Le dernier paramètre permettant de caractériser un réseau de neurones artificiels est le type d'apprentissage utilisé pour mettre à jour les poids synaptiques. Cette opération, aussi appelée *entraînement* du réseau, consiste à minimiser une fonction d'erreur représentant la différence entre la sortie provenant du réseau de neurones et le résultat attendu.

Plusieurs types d'apprentissage existent. Tout d'abord, l'apprentissage dit *supervisé* consiste à entraîner le réseau sur un ensemble de données de test dont le résultat est connu. Après avoir ajusté les poids synaptiques pour ces données de test, le réseau de neurones peut alors traiter d'autres données. Ce type d'apprentissage sert à calibrer des réseaux faisant de la classification binaire. L'apprentissage *non-supervisé*, quant à lui, consiste à traiter un ensemble de données de test considérées comme des variables aléatoires pour trouver une solution permettant des les séparer en

plusieurs catégories. Cet apprentissage est automatique et sert dans des applications de "*clustering*", consistant à diviser un ensemble de données en plusieurs catégories. Enfin, l'apprentissage *par renforcement* consiste à utiliser un agent neutre modifiant le réseau. Lorsqu'une partie du réseau est modifiée, l'environnement génère une fonction appelée *fonction de récompense*, positive ou négative. Le but de cet apprentissage est d'effectuer une série de modifications des poids synaptiques maximisant la fonction de récompense au cours du temps. Cet apprentissage est lui aussi automatique et est utilisé pour des applications dans le domaine de l'intelligence artificielle notamment.

Nous avons donc, dans un premier temps, introduit des concepts de bases pour la réalisation de réseaux de neurones artificiels. Dans la section suivante, nous présentons différents modèles de réseaux de neurones artificiels ainsi que plusieurs de leurs intégrations matérielles.

1.2 État de l'art des modèles de réseaux de neurones artificiels

Dans cette section, nous établissons un état de l'art des différents modèles de réseaux de neurones artificiels. Nous décrivons tout d'abord les deux principaux modèles : les réseaux de neurones à anticipation et les réseaux de neurones récurrents.

En considérant tous ces éléments, nous présentons finalement les verrous concernant l'intégration de ces modèles. Nous justifions le choix d'un modèle récent en regard des verrous qu'il élimine.

1.2.1 Présentation des types de réseaux de neurones artificiels

Les différents types de réseaux de neurones artificiels présents dans la littérature peuvent être classés en deux catégories. Ces catégories reprennent la classification selon l'organisation des neurones décrite dans la Section 1.1.1. D'un côté, les réseaux de neurones à anticipation incluent les réseaux de type *perceptron* et ses dérivés [Ros57]. Ce type de réseau sert de classificateur binaire entre une couche de neurones d'entrée et une couche de neurones de sortie. D'un autre côté, les réseaux de neurones récurrents incluent des réseaux tels que les réseaux de Hopfield [Hop82]. Ces derniers diffèrent des réseaux de neurones à anticipation par leur topologie. En effet, leurs neurones et connexions forment des cycles au sens de la théorie des graphes, c'est-à-dire il existe un chemin (ou boucle) d'un neurone vers lui-même dans le réseau. Les réseaux récurrents sont notamment utilisés en tant que mémoires associatives.

1.2.1.1 Réseaux de neurones à anticipation

Les réseaux de neurones à anticipation, aussi appelés réseaux de neurones de type "*Feedforward*", sont la plus ancienne forme de réseaux de neurones artificiels conçue [Ros57]. Ils consistent en un classificateur binaire entre des catégories, représentées par des neurones groupés en une *couche de sortie*. La plus simple version de ce type de réseau est le *perceptron simple couche* [Ros57]. Il n'est composé que d'une couche de sortie, sur laquelle les entrées sont directement fournies. Cependant, ce type de réseau n'est pas capable de traiter les problèmes non linéaires. Pour corriger cela, des couches

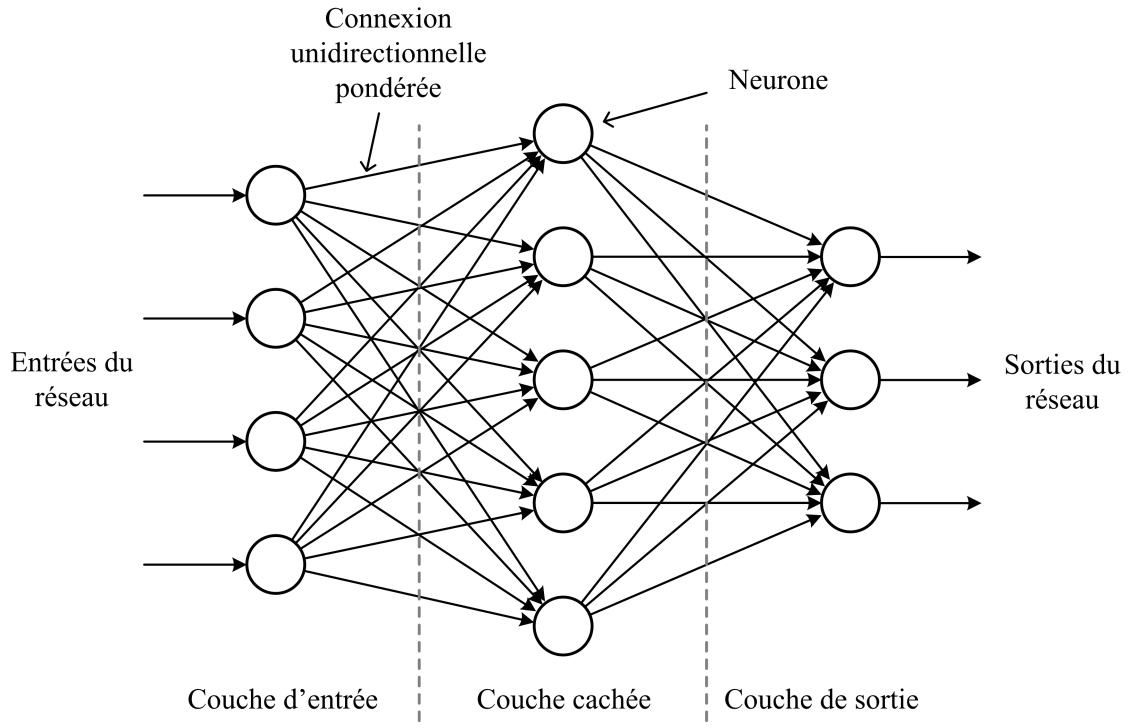

FIGURE 1.2 – Schéma de la structure d'un réseau de neurones à anticipation. Le réseau représenté est un réseau multi-couches comprenant une couche d'entrée, une couche cachée et une couche de sortie. Chaque couche est complètement connectée avec la couche suivante uniquement.

supplémentaires ont été ajoutées dans le réseau, formant le *perceptron multi-couches* [Ros61]. Les entrées sont fournies à des neurones formant une *couche d'entrée*. Une ou plusieurs couches appelées *couches cachées* sont insérées entre les couches d'entrée et de sortie, permettant un traitement plus fin des informations transitant dans le réseau. La Figure 1.2 montre un schéma de ce type de réseau avec une couche cachée. Dans un réseau à anticipation, une couche n'est connectée qu'à la couche suivante, par des liens unidirectionnels. Ainsi, les informations ne transitent que dans un seul sens.

Chaque connexion est pondérée par un poids synaptique. Ainsi, chaque neurone appartenant à une couche reçoit les informations pondérées de tous les neurones de la couche précédente. La décision sur l'activation ou l'inhibition d'un neurone est effectuée par une fonction d'activation sur la somme des données pondérées reçues. Cette fonction peut aller de la fonction d'Heaviside, ou fonction échelon unitaire, à la fonction sigmoïde. La complexité du neurone artificiel dépend donc de la complexité de la fonction d'activation et des poids synaptiques de ses connexions.

L'apprentissage dans un tel réseau est effectué à l'aide d'un algorithme appelé *rétro-propagation*. Ce dernier ajuste les poids synaptiques dans le but de minimiser la fonction d'erreur quadratique entre la sortie du réseau, c'est-à-dire les neurones de la couche de sortie, et le résultat espéré. Cet algorithme permet de faire un apprentissage supervisé dans le réseau.

D'autres réseaux de neurones dont le fonctionnement s'appuie sur les perceptrons ont aussi vu le

jour. Ainsi, les réseaux de neurones convolutionnels sont des variations de perceptrons multi-couches dont les neurones sont disposés de façon à reproduire le fonctionnement du cortex visuel [LBBH98]. Ils sont utilisés principalement dans des applications de reconnaissance d'images.

Ces réseaux ont l'avantage de garder une structure très simple tant que les problèmes auxquels ils sont confrontés sont linéaires. Dans le cas contraire, les fonctions d'activation se complexifient, pour aller par exemple jusqu'à la fonction sigmoïde. De plus, le stockage des poids synaptiques pour chaque connexion s'avère complexe à implanter. Pour une implantation numérique, chaque poids nécessite plusieurs bits de quantification, tandis qu'en électronique analogique, des composants passifs, coûteux en termes de surface, sont nécessaires, [MM14]-[BiPM04]. Des multiplications, coûteuses matériellement, doivent aussi être implantées pour chaque connexion. Le point faible de ce type de réseau est donc la complexité matérielle pour traiter des problèmes non linéaires.

1.2.1.2 Réseaux de neurones récurrents

Contrairement aux réseaux de neurones à anticipation, les réseaux de neurones récurrents ont des topologies permettant à un neurone d'envoyer son information à tous les autres neurones du réseau. Chaque neurone est donc connecté à tous les autres dans le réseau. Comme dans le cas des réseaux à anticipation, les connexions sont pondérées par des poids synaptiques pouvant être modifiés pendant le fonctionnement. Certains neurones peuvent être stimulés depuis l'extérieur du réseau et sont appelés *neurones d'entrée*. Les résultats sont récupérés en sortie de neurones appelés *neurones de sortie*. Les neurones restants sont des *neurones cachés*, et n'ont pas d'interaction avec l'extérieur du réseau. La Figure 1.3 montre le schéma d'un réseau récurrent contenant huit neurones. Parmi ceux-ci, trois sont des neurones d'entrée, deux sont des neurones de sortie, et les autres sont des neurones cachés. Tous les neurones sont connectés entre eux.

L'apprentissage dans un réseau récurrent modifie les poids synaptiques des connexions entre neurones. Typiquement, ceci est mis en place en suivant la *loi de Hebb* [Heb49], qui permet de faire un apprentissage par renforcement. Cette loi stipule que le poids synaptique entre deux neurones va augmenter si les deux neurones sont activés simultanément, et réduit si les deux neurones sont activés séparément.

Le plus célèbre type de réseau de neurones récurrents est le réseau de Hopfield [Hop82]. Il s'agit d'un cas particulier de réseau récurrent dans lequel tous les neurones servent de neurones d'entrée et de sortie. Ce type de réseau garantit une convergence vers un résultat, mais qui n'est pas toujours correct. Ce type de réseau est utilisé pour servir de mémoire associative, résistante à une altération des connexions.

Toutefois, le principal inconvénient de ce type de réseau est sa connectivité complète. En effet, le fait que chaque neurone soit connecté avec tous les autres dans le réseau donne une complexité des connexions quadratique en fonction du nombre de neurones. En considérant qu'un poids synaptique doit être associé à chaque connexion, l'implantation matérielle d'un réseau devient d'autant plus ardue que la taille du réseau augmente.

D'autres types de réseaux récurrents ont été développés pour pallier ce problème. Ainsi, les

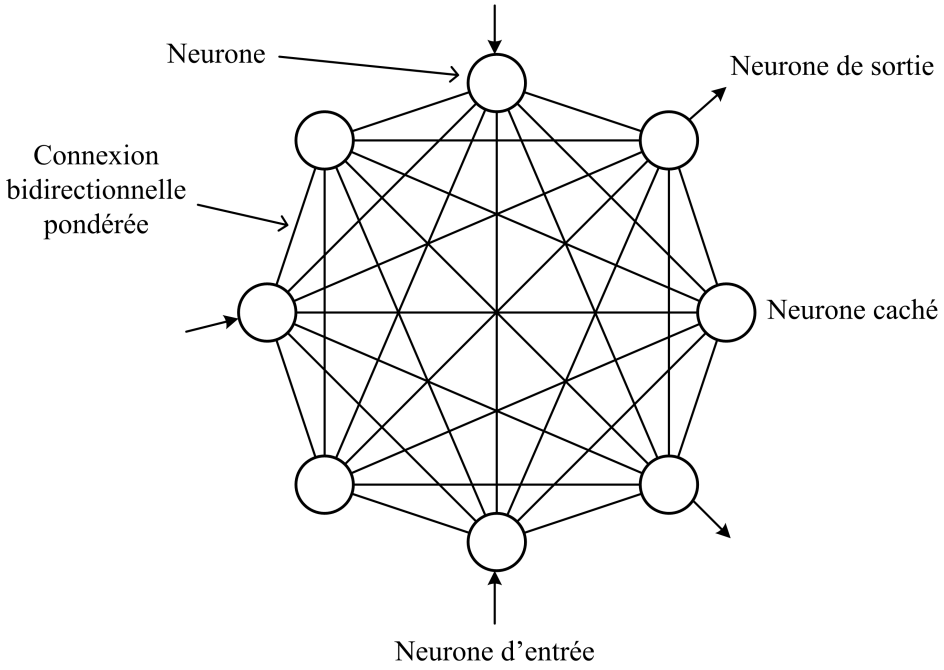

FIGURE 1.3 – Schéma de la structure d'un réseau de neurones récurrent contenant huit neurones. Parmi ceux-ci, trois sont des neurones d'entrée, deux sont des neurones de sortie, et les autres sont des neurones cachés. Tous les neurones sont connectés entre eux.

réseaux de Willshaw-Palm présentés dans [WBLH69] et [Pal13] permettent de ne pas effectuer toutes les connexions entre les neurones d'un réseau. D'abord, les motifs stockés dans le réseau sont composés d'un sous-ensemble de neurones seulement. Ensuite, seules les connexions entre neurones appartenant à des motifs stockés sont effectivement réalisées, ce qui réduit grandement le nombre total de connexions. Enfin, les poids synaptiques sont éliminés pour faire place à des connexions binaires. Cela simplifie davantage la réalisation matérielle. L'entraînement du réseau se transforme alors en stockage *a priori* des motifs grâce à des *cliques*. De plus, l'activation des neurones est faite grâce à des fonctions simples, comme la règle "Winner-Takes-All". Ce type de réseau présente aussi une capacité de stockage de motifs quadratique en fonction du nombre de neurones d'après [WBLH69], alors qu'elle est sous-linéaire dans un réseau de Hopfield.

Finalement, les principes des réseaux de Willshaw-Palm sont repris pour servir de base à un nouveau type de réseau : les réseaux de neurones à cliques [GB11]. La principale différence avec les réseaux de Willshaw-Palm est le groupement des neurones en plusieurs clusters de neurones. Cela permet de réduire les règles d'activation globales à l'échelle de ces groupements, entraînant une réduction supplémentaire de la complexité d'intégration pour une implantation matérielle.

1.2.2 Modèle adopté

Après avoir présenté plusieurs modèles de réseaux de neurones et les différentes façons de les intégrer sur un circuit, nous présentons maintenant nos choix sur le modèle de réseau de neurones artificiels à adopter. Notre but est de choisir un modèle simple permettant l'intégration de centaines, voire de milliers de neurones sur un circuit. En regard de ce que nous avons présenté dans la Section 1.2.1, de nombreux verrous rendent l'implantation d'un modèle complexe.

Dans les réseaux à anticipation, la connectivité complète entre deux couches et les poids synaptiques présents sur chaque connexion rendent une intégration matérielle difficile quand le nombre de neurones augmente. De plus, les poids synaptiques doivent être ajustables pour permettre la phase d'apprentissage dans le réseau. Enfin, la complexité de la fonction d'activation de chaque neurone, typiquement la fonction sigmoïde, est un frein à la réalisation d'un grand nombre de ces derniers. La complexité de la fonction d'activation de chaque neurone est en effet directement liée à la complexité globale du réseau.

Dans les réseaux de neurones récurrents classiques, comme les réseaux de Hopfield, le nombre de connexions entre neurones devient critique si tous les neurones doivent être connectés entre eux. De plus, comme pour les réseaux à anticipation, la présence d'un poids synaptique modifiable sur chaque connexion rend l'implantation matérielle d'un tel modèle complexe.

Ces verrous sont levés dans des réseaux ayant des connexions et des poids synaptiques binaires, comme les réseaux de Willshaw-Palm ou les réseaux de neurones à cliques décrits dans [GB11]. De plus, ces modèles utilisent des fonctions d'activation simples nécessitant seulement une comparaison du nombre de contributions, comme la règle d'activation "Winner-takes-All". Ces modèles sont donc adaptés à la réalisation matérielle de réseaux de grande taille (centaines voire milliers de neurones).

Cependant, dans un réseau de neurones à cliques décrit dans [GB11], le réseau est divisé en plusieurs groupes de neurones. Contrairement aux réseaux de Willshaw-Palm, les fonctions d'activation peuvent ainsi être locales à un groupement de neurones, ce qui simplifie l'intégration du réseau.

Afin de réduire la complexité d'un réseau de neurones artificiels au maximum, nous utilisons donc le modèle des réseaux de neurones à cliques décrits dans [GB11].

Dans la section suivante, nous présentons plus en détail le modèle des réseaux de neurones à cliques et les opérations que nous devons implanter sur circuit électronique.

1.3 Description des réseaux de neurones à cliques

Dans cette section, nous décrivons un nouveau type de réseaux de neurones : les réseaux de neurones à cliques [GB11]. Ce modèle s'inspire des réseaux de Willshaw-Palm décrits dans [WBLH69] et [Pal13]. Un réseau de neurones à cliques est composé de $N_{F_{tot}}$ neurones, appelés *fanaux* dans [GB11]. Ces derniers sont connectés entre eux par des liens binaires afin de former des *cliques* stockant des *messages* d'information dans le réseau. Nous décrivons dans cette section la

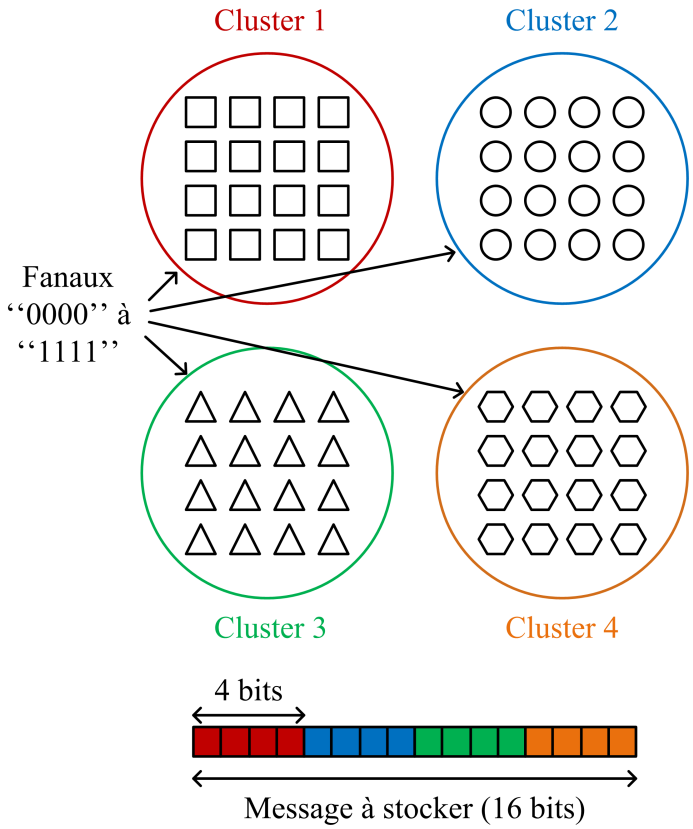

FIGURE 1.4 – Schéma de l'organisation des fanaux dans un réseau à cliques. Cet exemple montre un réseau de quatre clusters contenant chacun seize fanaux. Ce réseau est destiné à stocker des mots binaires de seize bits.

structure du réseau et d'un fanal. Puis, nous expliquons les procédure de stockage et de décodage de messages dans le réseau.

1.3.1 Structure du réseau et description des opérations

Dans un réseau de neurones à cliques, les fanaux sont organisés en N_C groupes disjoints appelés *clusters*, comme dans la Figure 1.4. Un cluster représente une catégorie d'information dans laquelle chaque fanal est une partie de l'information liée à cette catégorie. Chaque cluster contient donc $N_{F_{tot}} \div N_C = N_F$ fanaux, et chaque fanal porte l'information de $\log_2(N_F) = N_{bits}$ bits d'information. Dans l'exemple de la Figure 1.4, le réseau représenté est conçu pour stocker des mots de seize bits. Il est composé de quatre clusters de seize fanaux chacun. Chaque cluster représente donc une sous-partie de quatre bits des messages à stocker. Les seize fanaux appartenant à chaque cluster servent donc à coder tous les états que peut prendre le cluster.

Les fanaux sont connectés entre eux par des liaisons binaires d'un cluster vers un autre, c'est-à-dire qu'une connexion entre deux fanaux appartenant à des clusters différents peut exister ou non. Si cette connexion existe, contrairement à d'autres modèles de réseaux de neurones, elle n'est pas

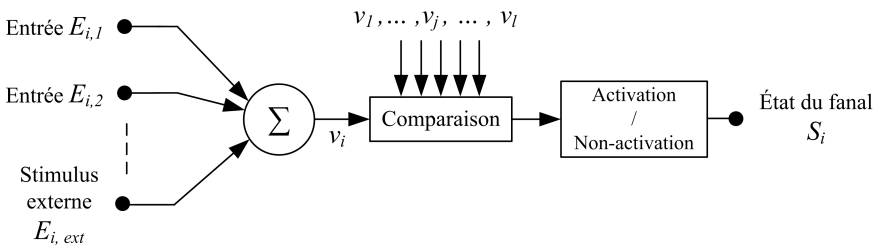

FIGURE 1.5 – Schéma d'un fanal selon le modèle de [GB11].

pondérée par un poids synaptique.

La composition d'un fanal, par exemple le fanal $\#i$ dans un cluster, ayant n entrées provenant d'autres fanaux est donnée sur la Figure 1.5. Une entrée de stimulation externe est également présente dans le fanal. Toutes ces contributions binaires – provenant d'autres fanaux et de l'extérieur – sont ajoutées pour donner un résultat ν_i, aussi appelé *nombre de contributions actives* :

$$\forall i, 1 \leq i \leq N_F : \quad \nu_i = \sum_{j=1}^{n} E_{i,j} + E_{i,ext}, \tag{1.1}$$

où $E_{i,j}$ est la valeur de l'entrée $\#j$ du fanal $\#i$ et E_{ext_i} la valeur de l'entrée de stimulation externe de ce fanal. Le résultat ν_i de l'addition est ensuite comparé avec les ν_j provenant d'autres fanaux. Si le résultat ν_i du fanal $\#i$ respecte la règle d'activation dépendant du résultat de la comparaison, sa sortie S_i passe à '1' et l'état de ce fanal est dit *actif*.

Plusieurs règles d'activation, définies dans [AGJ14], peuvent être utilisées :

- la règle d'activation "Winner-Takes-All" ;
- la règle d'activation "k-Winners-Take-All" ;
- la règle d'activation "k-Losers-Kicked-Out".

Le choix de l'une de ces règles influe sur la méthode de comparaison et le nombre de résultats ν_j auquel un résultat ν_i est comparé. Nous décrirons ces différentes règles d'activation dans la Section 1.3.3.2.

1.3.2 Stockage de messages dans le réseau

L'ensemble des messages stockés dans le réseau à cliques est appelé *dictionnaire* du réseau. Les messages sont stockés *hors ligne*, c'est-à-dire avant de mettre le réseau en fonctionnement, et représentent ainsi la connaissance *a priori* du réseau. Les messages sont constitués de plusieurs sous-parties, chacune d'elles représentée par un cluster. Ainsi, dans l'exemple de la Figure 1.4, les quatre premiers bits des messages sont représentés par les fanaux du cluster rouge, les quatre suivants par ceux du cluster bleu, et ainsi de suite. Dans l'exemple, un message particulier sera donc représenté par quatre fanaux, un par cluster.

Pour stocker un message dans le réseau, les fanaux le représentant sont interconnectés de façon

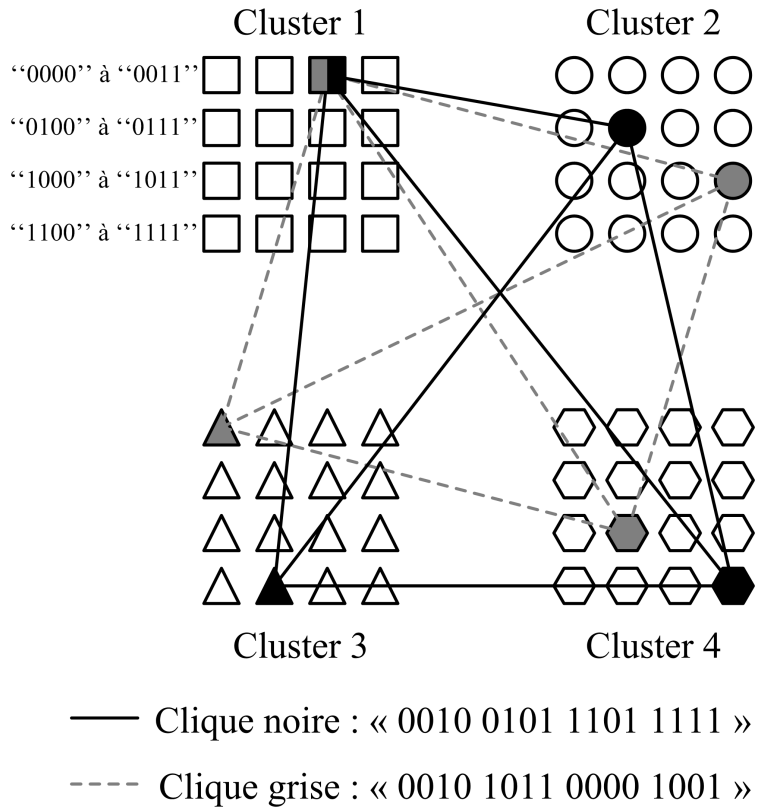

FIGURE 1.6 – Schéma d'un réseau à cliques dans lequel deux messages sont stockés, un représenté par la clique noire, l'autre par la clique grise.

complète. Ils forment ainsi un sous-graphe dans le réseau appelé *clique*. La conséquence directe de ce mode de stockage est que deux fanaux appartenant à un même cluster ne peuvent pas être connectés ensemble. La Figure 1.6 montre l'exemple de deux cliques – et donc deux messages stockés – réalisées dans le réseau de la Figure 1.4. Cette terminologie, utilisée dans [GB11], est aussi utilisée en neurobiologie pour décrire de telles assemblée de neurones, comme dans [LOZ06].

Lors du stockage de plusieurs messages, il peut arriver que plusieurs cliques partagent la même connexion. Dans ce cas, cette connexion n'est pas dupliquée, elle reste inchangée. L'ordre dans lequel les messages sont stockés dans le réseau n'a donc pas d'importance.

Le nombre de sommets d'une clique n'est pas nécessairement égal à N_C. Les messages représentés par ces cliques ne sont donc pas représentés par un fanal de chaque cluster, et sont appelés *messages parcimonieux*. Un plus grand nombre de ces messages peut être stockés, mais leur récupération nécessite des fonctions réalisées à l'échelle du réseau entier. [ABGJ14] décrit plus en détail le stockage et la récupération de messages parcimonieux. Pour simplifier notre étude, nous ne considérons pas le cas de messages parcimonieux.

La *densité d* d'un réseau à cliques est définie comme le rapport entre le nombre de connexions réalisées et le nombre total de connexions possibles. Ce paramètre est utilisé pour représenter le

nombre de messages stockés dans le réseau en s'affranchissant des paramètres de taille du réseau. En effet, un fanal appartenant à une clique est relié à $(N_C - 1)$ autres fanaux. Comme une clique comporte N_C fanaux, et en supposant un stockage des messages uniforme sur l'ensemble du réseau, le nombre de connexions bidirectionnelles réalisées devient :

$$N_{conn\_eff} \approx \frac{N_{messages} N_C (N_C - 1)}{2}. \tag{1.2}$$

Le nombre de connexions total pouvant être réalisé est égal au nombre de connexions incidentes à un fanal, multiplié par le nombre de fanaux total $N_{F_{tot}}$. Sachant qu'un fanal peut recevoir des connexions venant de tous les autres fanaux des autres clusters, un fanal peut recevoir jusqu'à $N_F(N_C - 1)$ connexions. Le nombre total de connexions bidirectionnelles dans le réseau est donc :

$$N_{conn\_tot} = \frac{N_{F_{tot}} N_F (N_C - 1)}{2}. \tag{1.3}$$

Ainsi, grâce aux équations (1.2) et (1.3), d peut être assimilé au ratio du nombre de messages stockés dans le réseau $N_{messages}$ et du nombre de fanaux par cluster au carré :

$$\begin{aligned} d &\approx \frac{2 N_{messages} N_C (N_C - 1)}{2 N_{F_{tot}} N_F (N_C - 1)} \\ &\approx \frac{N_{messages}}{N_F^2}. \end{aligned} \tag{1.4}$$

On peut donc calculer le nombre de bits pouvant être stockés dans le réseau en fonction de la densité et des paramètres de taille du réseau :

$$\begin{aligned} N_{bits} &= N_{messages} N_C \log_2(N_F) \\ &\approx d N_F^2 N_C \log_2(N_F). \end{aligned} \tag{1.5}$$

En revanche, il existe une limite au nombre de cliques pouvant être réalisées dans un réseau à cliques. En effet, plus le nombre de connexions dans le réseau augmente, plus le risque de créer des "cliques parasites", c'est-à-dire des cliques réalisées mais non désirées, existe. La récupération des messages n'est alors plus fiable. D'après [GB11], la limite de densité empirique qu'un réseau à cliques peut accepter sans souffrir de perte de performance est d'environ 25% pour un répartition des messages aléatoire et uniforme.

Le nombre de bits maximum pouvant être stockés dans un réseau devient donc :

$$N_{bits\_max} \approx \frac{1}{4} N_F^2 N_C \log_2(N_F)$$
$$\approx \frac{1}{4} \left(\frac{N_{F_{tot}}^2}{N_C} \right) \log_2 \left(\frac{N_{F_{tot}}}{N_C} \right). \quad (1.6)$$

Il est intéressant de remarquer que, pour un nombre de fanaux total égal, plus de bits peuvent être stockés dans un réseau comportant moins de clusters. Les cliques ont alors moins de sommets, mais elles sont plus nombreuses et chaque fanal est équivalent à un plus grand nombre de bits.

1.3.3 Récupération de messages

1.3.3.1 Processus de récupération

La récupération d'un message consiste à retrouver un message appris dans le dictionnaire en le connaissant entièrement ou en partie seulement. La rétroaction inhérente aux cliques apporte une forte corrélation entre les différentes parties du message à retrouver. Ainsi, cela permet de compenser une ou des altérations dans la stimulation initiale.

Le processus de récupération est organisé comme suit. Tout d'abord, les fanaux représentant le message présenté en entrée du réseau sont stimulés. Les fanaux échangent ensuite les informations sur leur état entre eux, tout en mettant à jour leurs entrées provenant d'autres fanaux. Ce processus est itéré N_{IT} fois, jusqu'à ce que le réseau entre dans un état stable, c'est-à-dire les états des fanaux ne changent plus. D'après [GB11], quatre itérations sont suffisantes pour arriver à un résultat stable dans tous les cas. Les fanaux actifs à la fin de la récupération forment alors le message récupéré et éventuellement corrigé par le réseau.

L'algorithme correspondant à ce processus pour N_C clusters contenant chacun N_F fanaux est le suivant :

Stocker hors ligne le contenu du dictionnaire dans le réseau.
Tant que en ligne
  Initialiser de l'état des fanaux et de leurs entrées.
  Stimuler les fanaux correspondant au message d'entrée.
  Pour tout itération $k = 1$ à N_{IT}
    Pour tout cluster $j = 1$ à N_C
      Pour tout fanal $i = 1$ à N_F
      Additionner les contributions des entrées = ν_i.
      Stocker ν_i.
      Fin
    Fin
  Comparer le nombre des contributions actives des fanaux.
  Sélectionner les fanaux respectant la règle d'activation.
  Mettre l'état des fanaux sélectionnés à '1' et celui des autres à '0'.
  Propager les états des fanaux vers les entrées des fanaux connectés.
  Mettre à jour les entrées des fanaux.
  Fin
Fin

La complexité de cet algorithme dépend en grande partie de la règle d'activation. Cette dernière influe en effet sur le nombre d'éléments mis en jeu dans l'étape de comparaison, et représente l'essentiel des calculs effectués dans un fanal.

1.3.3.2 Choix de la règle d'activation

1.3.3.2.1 Règle d'activation "Winner-Takes-All"

La règle d'activation "Winner-Takes-All" (WTA) consiste à activer, dans chaque cluster, le fanal ayant le plus grand nombre de contributions actives. Cette règle est donc locale à un cluster et peut être définie par l'équation suivante :

$$\forall i, 1 \leq i \leq N_F : S_i = \begin{cases} 1 & \text{si} \quad \nu_i = \max\limits_{1 \leq j \leq N_F} (\nu_j), \\ 0 & \text{sinon.} \end{cases} \tag{1.7}$$

Le fait que cette opération soit effectuée localement simplifie l'opération de comparaison des résultats ν_j, qui peut être effectuée sur un plus petit nombre d'éléments que pour une règle globale. De plus, l'opération de comparaison est d'autant plus simplifiée que seuls les fanaux ayant le plus de contributions sont recherchés. Les autres fanaux n'ont pas besoin d'être classés entre eux, et la comparaison se résume donc à la recherche d'un maximum.

Le principal inconvénient de la règle d'activation WTA locale est qu'elle ne convient pas à la

récupération de messages parcimonieux. Comme ces derniers n'utilisent pas tous les clusters, il est nécessaire de différencier au niveau du réseau les fanaux appartenant à la clique recherchée des autres. Or, la règle d'activation WTA locale ne permet pas de faire cette distinction.

1.3.3.2.2 Règle d'activation "k-Winners-Take-All"

La règle d'activation "k-Winners-Take-All" (WsTA) consiste à n'activer, dans l'ensemble du réseau, que les k fanaux ayant le plus de contributions actives. Cette règle est, cette fois, globale et peut être décrite par l'équation suivante :

$$\forall i, 1 \leq i \leq N_{F_{tot}} : S_i = \begin{cases} 1 & \text{si} \quad \nu_i \geq \nu_k, \\ 0 & \text{sinon.} \end{cases} \tag{1.8}$$

où ν_k est le résultat d'addition du k-ième fanal ayant le *plus* de contributions actives.

Contrairement à la règle d'activation WTA locale, la règle d'activation WsTA nécessite de comparer et de classer tous les fanaux du réseau entier en fonction de leur nombre de contributions actives. Cela complexifie la fonction de comparaison par rapport à celle réalisant la règle WTA locale.

En revanche, cette règle est utilisée dans le cas de stockage de messages parcimonieux. Dans ce cas, le paramètre k est fixé de manière à être égal au nombre de sommets de la clique recherchée, même en cas d'effacements sur les entrées. Cela implique de connaître, à tout instant, le nombre de sommets de la clique recherchée.

1.3.3.2.3 Règle d'activation "k-Losers-Kicked-Out"

La règle d'activation "k-Losers-Kicked-Out" (LsKO) consiste, à partir d'un ensemble de fanaux activés dans le réseau, à éliminer définitivement les k fanaux actifs ayant les plus faibles nombres de contributions actives. Ce processus est répété jusqu'à ce que tous les fanaux actifs restants aient le même nombre de contributions actives. Cette règle est donc une règle globale.

Ainsi, après la stimulation et une première phase de propagation des états, l'activation est faite par une règle WTA globale, afin de déterminer l'ensemble de fanaux activés au départ. Les phases d'activation suivantes peuvent être formalisées par l'équation suivante :

$$\forall i, 1 \leq i \leq N_{F_{tot}} : S_i = \begin{cases} 1 & \text{si} \quad \nu_i \geq \nu_k, \\ 0 & \text{jusqu'à la fin du décodage sinon.} \end{cases} \tag{1.9}$$

où ν_k est le résultat d'addition du k-ième fanal ayant le *moins* de contributions actives.

Par rapport à la règle WsTA, la règle d'activation LsKO a l'avantage de ne pas avoir besoin de connaître le nombre de sommets de la clique recherchée. Les fanaux sont éliminés petit à petit jusqu'à converger vers une clique.

En revanche, contrairement aux règles précédentes, le comportement de cette règle a plusieurs

phases. Il varie donc en fonction de l'avancement de la récupération du message. Cela signifie que la récupération ne peut pas s'effectuer de manière continue et qu'elle nécessite une supervision extérieure au réseau.

1.4 Intégration matérielle des réseaux de neurones à cliques

Dans cette section, nous exposons deux modèles d'implantation d'un neurone. D'abord, nous montrons des circuits utilisant une approche neuromimétique, dont le but est de reproduire fidèlement le comportement des neurones biologiques. Puis, nous décrivons des circuits bio-inspirés, s'affranchissant du support biologique pour s'intéresser au comportement des neurones au niveau informationnel.

Nous faisons finalement un choix sur le modèle que nous allons adopter pour l'implantation d'un neurone.

1.4.1 Circuits neuromimétiques

La famille des circuits neuromimétiques a pour objectif de mimer le plus exactement possible le comportement des neurones biologiques. Ces derniers communiquent grâce à la génération d'impulsions appelées *potentiels d'action*, [Mea89]-[MB99]. Plusieurs modèles biologiques sont adoptés dans la littérature pour la génération des potentiels d'actions, plus ou moins fidèles au modèle biologique.

Tout d'abord, le modèle *"Integrate-and-Fire"* (IF) est le plus simple et un des plus vieux de ces modèles [Abb99]. Le neurone est ainsi représenté comme une capacité – celle de sa membrane – chargée par un courant d'entrée. Ce courant d'entrée représente la somme des entrées pondérées par les synapses ("Integrate"). La tension aux bornes de cette capacité va ainsi augmenter, jusqu'à ce qu'elle dépasse une tension de seuil constante, dépendante de la loi d'activation du neurone. Le neurone émet alors un potentiel d'action et la tension aux bornes de la capacité diminue à sa valeur de repos ("Fire"). Ainsi, pour une valeur continue du courant d'entrée, ce dernier est lié linéairement à la fréquence d'émission du potentiel d'action. Il n'y a pas de phénomène de saturation à cause d'une trop grande valeur du courant d'entrée.

Ce modèle est utilisé dans des réseaux de type *Feedforward*, par exemple par Hsieh *et al.* qui s'inspirent du système olfactif [HT12], ou par Bofill-i-Petit *et al.* dans [BiPM04] pour de la détection de synchronisme. Il est aussi utilisé dans des réseaux de neurones récurrents, comme dans [ICD06], [CSBI14] ou [CGD$^+$07].

Cependant, le modèle IF n'est pas tout à fait conforme au comportement d'un neurone biologique. En effet, dans le modèle IF, si un courant faible arrive en entrée, la capacité va tout de même se charger et garder les charges ainsi acquises dans le temps. Ce n'est pas le cas dans un neurone biologique, qui n'émet son potentiel d'action que si un courant minimal se présente à sa membrane. Le modèle IF est donc modifié pour prendre en compte ce comportement, et devient le modèle

"Leaky-Integrate-and-Fire" (LIF). Ainsi, la capacité du neurone ne se charge que si le courant en entrée du neurone dépasse un courant de seuil constant.

Des implantations analogique et numérique de neurones LIF sont proposées par Joubert *et al.* dans [JBTH12]. De plus, ce type de neurone est aussi intégré par Corradi *et al.* dans un réseau récurrent dans [CZR+14] afin d'émuler le comportement du système vestibulaire du cerveau.

Enfin, des modèles sont inspirés d'expériences réalisées sur de vrais neurones. C'est le cas du modèle de neurone de Hodgkin-Huxley (HH), développé à partir de tests réalisés sur des neurones de calamar [HH52]. Ce modèle est celui qui se rapproche le plus du comportement du neurone biologique et un des modèles les plus couramment utilisés. Cependant, il est aussi un des plus complexes. L'émission du potentiel d'action est en effet conditionnée à un ensemble d'équations différentielles non-linéaires, représentant le comportement électriques des composants chimiques présents dans le neurone. Des opérateurs analogiques sont présentés par Saïghi *et al.* dans [SBT+11] afin de concevoir des neurones de ce type. De plus, un réseau récurrent de quatre neurones HH est implanté par Yu *et al.* dans [YC10]. Des versions simplifiées de ce neurone ont été proposées et implantées en circuit analogique, comme le modèle de FitzHugh-Nagumo [Fit61]-[NAY62], ou encore par Wijekoon *et al.* [WD08].

Il existe aussi des réseaux de neurones neuromimétiques entièrement configurables. C'est le cas par exemple de SpiNNaker [PPG+13], conçu entièrement à l'aide de l'électronique numérique à base de processeurs ARM®, et de TrueNorth [Mer14], conçu lui aussi de manière numérique. Dans ces deux implantations, les potentiels d'action sont générés sous la forme de trames de données numériques, circulant dans une architecture de type "Network-on-Chip" (NoC). Cela rend l'organisation du réseau flexible (réseau *Feedforward* ou récurrent), et la génération de ces potentiels d'action peut suivre plusieurs modèles (LIF, HH, etc.).

Hormis ce dernier exemple, les réseaux neuromimétiques n'implantent que des réseaux de l'ordre d'une dizaine de neurones au maximum. Ils ne sont donc pas destinés à l'intégration d'un grand nombre de neurones sur un même circuit.

1.4.2 Circuits bio-inspirés

Afin de simplifier les circuits implantant des réseaux de neurones artificiels, l'approche bio-inspirée permet de réaliser les échanges d'informations dans les neurones sans tenir compte des potentiels d'action. Ainsi, les courants et tensions dans un circuit représentent directement les signaux traités dans les différents modèles. Avec ces contraintes en moins, les neurones ont alors la possibilité de réagir plus rapidement qu'à l'échelle biologique. L'objectif de ces circuits est donc d'être plus efficaces en termes de temps de réaction et de surface.

De nombreux circuits utilisent cette approche pour intégrer les modèles de réseaux précédemment décrits. Ainsi, dans [CFSV11], Carvajal *et al.* intègrent un perceptron mono-couche à 32 entrées résistant aux bruits environnants ou au défaut d'appariement des transistors. Des perceptrons multi-couches ont aussi été intégrés par Gatet *et al.* [GTBB09], Maliuk *et al.* [MM14] ou encore Milev *et al.* [MH03].

Les réseaux récurrents sont aussi intégrés en considérant cette approche. Par exemple, Kier *et al.* implantent un réseau récurrent comportant quatre neurones sur puce [KABH06], tandis qu'un réseau de Hopfield de 25 neurones est implanté sur FPGA par Maeda *et al.* [MF07].

Cette approche est de plus utilisée quand le réseau de neurones est intégré dans un système complet, quand le résultat est utilisé par une autre fonction en aval. Des circuits plus simples peuvent alors permettre d'intégrer un plus grand nombre de neurones dans un réseau.

1.4.3 Modèle de neurone adopté

Nous choisissons d'intégrer le modèle des réseaux de neurones à cliques de manière bio-inspirée. Si l'on choisit d'implanter un réseau de plusieurs centaines, voire des milliers de neurones, il est en effet nécessaire de réduire davantage la complexité des traitements effectués. Nous réalisons cela en nous passant des potentiels d'actions. De plus, notre priorité n'est pas d'imiter au plus près le comportement d'un neurone biologique, mais d'utiliser les réseaux de neurones en tant qu'unité de calcul. Pour toutes ces raisons, la méthode d'intégration neuromimétique ne convient pas à notre problématique.

Cependant, l'intégration d'un circuit comportant un grand nombre de neurones (ou fanaux) peut poser plusieurs problèmes. La surface du circuit doit garder un degré de dépendance minimal par rapport au nombre de neurones (ou fanaux). Si ce n'est pas le cas, le circuit devient trop encombrant lorsque ce nombre augmente, et l'intérêt de faire un circuit dédié est perdu. La consommation d'énergie du circuit est aussi importante dans un contexte d'utilisation portable, c'est-à-dire alimenté par batterie.

Conclusion

Dans ce chapitre, nous avons fait l'inventaire des différents modèles de réseaux de neurones artificiels et des méthodes d'intégration de ces modèles sur circuit. Nous avons donc choisi d'explorer l'intégration du modèle des réseaux de neurones à cliques, que nous avons décrit en détail. Ce modèle de réseau permet de diminuer la complexité des connexions présente dans les autres types de réseaux de neurones récurrents, ou entre les couches de réseaux de neurones de type *Feedforward*. Les poids synaptiques sont également absents du modèle des réseaux de neurones à cliques, simplifiant davantage le modèle.

En outre, afin de pouvoir intégrer un grand nombre de fanaux, nous avons choisi l'approche bio-inspirée. Elle ne cherche pas à reproduire le comportement exact des neurones biologiques mais se concentre sur le comportement des neurones au niveau informationnel. Cette approche permet d'utiliser des fonctions plus simples afin de réduire la complexité d'intégration du modèle. Une intégration sur ASIC semble donc prometteuse pour des applications demandant la puissance de calcul des réseaux de neurones tout en conservant une consommation de ressources et d'énergie faible.

Dans le prochain chapitre, nous proposons des organisations génériques des éléments du réseau permettant d'intégrer un nombre indéterminé de fanaux dans un réseau à cliques.

Chapitre 2

Architectures mixtes analogiques/numériques d'un réseau de neurones à cliques

Introduction

Le chapitre précédent a présenté les principes des réseaux de neurones à cliques, et comment ils se positionnent dans l'état de l'art. Les réseaux à cliques simplifient les opérations dans les cœurs de calculs et se basent sur des connexions binaires connectant uniquement les nœuds corrélés entre eux. Ainsi, il a été montré qu'une plus grande quantité d'information pouvait être stockée dans un réseau à cliques que dans un réseau de Hopfield de même taille, par exemple. Dans ce chapitre, nous décrivons les choix que nous adoptons sur la technologie pour l'intégration du circuit. Nous avons pour but de proposer des organisations génériques des éléments du réseau, permettant de connecter un nombre indéterminé de fanaux. Nous étudions enfin, pour chaque organisation proposée, le degré de dépendance de sa complexité en fonction du nombre de fanaux, en termes de surface de silicium occupée et de latence de traitement.

2.1 Réflexions préalables à l'intégration

Il existe de nombreuses manières d'implanter un système sur circuit. La technologie utilisée, le type d'implantation (analogique ou numérique), le support d'intégration (FPGA ou ASIC), sont autant de choix offerts pour la conception d'un circuit intégré.

Dans cette section, nous discutons des choix effectués avant la conception du circuit implantant les réseaux de neurones à cliques. Nous décrivons tout d'abord quels sont les critères sur lesquels devront se concentrer nos efforts durant la conception du circuit. Ensuite, nous justifions l'intérêt de l'utilisation d'une conception mixte analogique/numérique par rapport à d'autres circuits implantant les réseaux de neurones à cliques. Enfin, nous expliquons notre choix de la technologie Si CMOS 65 nm.

2.1.1 Critères d'évaluation d'un circuit

Les performances d'un circuit intégré sont typiquement évaluées selon quatre critères :

- la surface de semi-conducteur ;
- la latence ;
- le débit de traitement ;
- la puissance consommée.

La surface de semi-conducteur occupée par un circuit intégré est un paramètre important à maîtriser. De plus, la dépendance de cette surface par rapport au nombre d'éléments devient cruciale lors de l'intégration d'un grand nombre d'éléments, comme dans notre cas. Le circuit doit en effet rester le plus compact possible lorsque le nombre d'éléments à intégrer augmente, notamment dans une application portable par exemple, et pour des raisons de coûts de fabrication. Nous devrons donc étudier cette dépendance au nombre d'éléments dans la conception du circuit, et la minimiser si possible.

La latence du circuit est ensuite définie comme la durée entre l'instant où sont présentés les signaux à l'entrée du circuit et l'instant où les sorties du circuit sont disponibles. Comme expliqué dans le Chapitre 1, nous ne nous plaçons pas dans une approche neuromimétique, nous n'avons pas d'objectif en termes de latence, si ce n'est d'en connaître les limites en fonction de nos choix d'intégration. Il est aussi possible de réutiliser du matériel pour diminuer la surface de circuit, au prix d'une plus grande latence.

Le débit de traitement représente la quantité d'information (le nombre de bits) traitée par le circuit par unité de temps. Dans une application en temps réel, ce critère conditionne le fonctionnement du circuit. Or, comme pour la latence, nous n'avons pas d'objectif particulier pour le débit, mais nous cherchons à en connaître les limites également.

Enfin, nous considérons aussi la puissance consommée dans un circuit intégré. C'est un paramètre que nous cherchons à minimiser, à plus forte raison dans le cas d'une utilisation portable, c'est-à-dire alimenté par batterie. On peut aussi parler de consommation d'énergie, en considérant la combinaison de la puissance consommée, de la latence et du débit.

2.1.2 Les techniques d'intégration et les supports associés

Une implantation matérielle d'un circuit peut être réalisée de plusieurs manières : à l'aide de l'électronique analogique, de l'électronique numérique ou d'un mélange des deux (on parle alors d'implantation mixte analogique/numérique). Les implantations de fonctions analogiques permettent d'utiliser les caractéristiques internes des composants de l'électronique (transistors, résistances, capacités, inductances, etc.) et les lois de Kirchhoff pour réaliser des opérations en apparence complexes sur des circuits de faible complexité et consommant peu de puissance. Par exemple, les additions lorsque les grandeurs à traiter sont des courants sont réalisées par de simples nœuds. En revanche, ces circuits sont vulnérables à plusieurs types d'imperfections dues à l'intégration sur circuit, comme le désappariement de transistors ou les variations de paramètres environnementaux, comme la tension d'alimentation ou la température par exemple. Ainsi, dans les circuits analogiques, les composants, notamment les transistors de taille minimale ne sont pas utilisés pour diminuer l'impact de ces imperfections.

Les implantations analogiques de circuits se font sur des circuits dédiés à l'application pour laquelle ils vont être utilisés, les ASICs. Cela impose une conception des masques du circuit intégré dite *"full-custom"*, c'est-à-dire que chaque composant est placé sur le circuit et connecté aux autres par le concepteur. Les composants à placer sur le circuit sont issus d'un schéma électrique du circuit.

Les fonctions numériques reposent, elles, sur l'utilisation d'informations binaires. Cela leur confère une grande résistance aux bruits dans les échanges d'informations et permet d'utiliser des transistors de taille minimale à l'intérieur des fonctions. Par contre, le traitement de données provenant du monde extérieur, intrinsèquement analogique, nécessite une étape de conversion vers des informations binaires. De plus, le traitement numérique s'appuie sur un nombre restreint d'opérateurs élémentaires issus des logiques combinatoires (algèbre de Boole) et séquentielles (nécessitant des unités de mémoire). Chaque fonction est donc une composition de ces opérateurs élémentaires, et est donc en général plus complexe en nombre d'éléments qu'un équivalent analogique.

Les implantations numériques, quant à elles, se font à partir de la description du circuit grâce à un langage comme le VHDL ou le Verilog. Cette description représente l'architecture et le comportement du circuit. Elle est ensuite convertie vers un schéma composé de portes logiques par un procédé appelé *"synthèse"*. Ce schéma peut ensuite être intégré sur un ASIC en utilisant des composants issus d'une bibliothèque de portes logiques, ou intégré sur un circuit FPGA (*Field Programmable Gate Array*). Un circuit FPGA est un circuit reconfigurable composé de multiples blocs logiques élémentaires (de quelques milliers à quelques millions) pouvant s'interconnecter. Chaque bloc logique contient en général des points mémoires et des LUTs (*Look-Up Table*), éléments programmables servant à implémenter des équations logiques à plusieurs entrées. Il est donc possible de changer ce qui se trouve sur le circuit en changeant les connexions entre blocs et la configuration des fonctions logiques implémentées, ce qui n'est pas le cas avec un ASIC. Ceci mis à part, les ASIC ont des meilleures performances en termes de latence et de consommation, car ils sont optimisés pour l'application qui les utilise.

Enfin, une fois le support choisi, il est possible de choisir plusieurs technologies d'intégration. Pour un ASIC, il est possible de choisir le type de substrat (silicium en immense majorité, autres substrats en silicium-germanium), le type de transistors (bipolaire ou MOS) ainsi que la finesse de gravure de ces derniers. Ce dernier paramètre impacte non seulement la surface du circuit, mais aussi la latence et la consommation car des transistors plus petits ont de plus faibles capacités parasites et permettent d'utiliser de plus faibles courants. En revanche, la proportion de courants de fuite est plus importante dans des transistors plus petits. Actuellement, les ASIC analogiques implantant des réseaux de neurones utilisent des finesses de gravure comme 0,35 μm dans [CFSV11] ou [MM14], ou 0,18 μm dans [HT12], qui sont des technologies peu récentes (mises en circulation en 1995 et 1999) mais fiables. Les ASIC numériques implantant des réseaux de neurones utilisent des technologies plus récentes, ayant des finesses de gravure allant jusqu'à 28 nm [Mer14]. Pour un circuit implanté sur FPGA, cela dépend du circuit FPGA utilisé. Le choix du circuit FPGA va alors avoir un effet sur la quantité de blocs logiques disponibles, ainsi que la latence du circuit selon la finesse de gravure du circuit FPGA lui-même.

2.1.3 Intégrations existantes des réseaux à cliques

À notre connaissance, il existe deux implantations circuit des réseaux de neurones à cliques issus de [GB11]. Ces deux circuits sont des implantations des réseaux à cliques utilisant des fonctions d'électronique numérique. Le modèle des réseaux de neurones à cliques décrit dans [GB11] se basant sur des informations binaires, il n'y a donc pas besoin de convertir le format de données pour effectuer les calculs dans le fanal. En revanche, les approches utilisées pour concevoir chacun des deux circuits sont assez différentes.

Pour commencer, Jarollahi *et al.* [JOGG14] et nous-mêmes [LLAS13] suivons un flot de conception de circuit numérique proche de celui décrit dans la Section 2.1.2. En partant de la description algorithmique du modèle décrite dans la Section 1.3.3.1, un modèle bas niveau du réseau est développé pour servir de référence au circuit numérique. Ensuite, la synthèse du circuit est réalisée en passant par un langage de description du circuit, en partant du modèle bas niveau. Le circuit est finalement implanté sur FPGA. L'utilisation de l'électronique numérique permet au circuit de réutiliser du matériel plusieurs fois pour diminuer la surface de silicium utilisée, au prix d'une plus grande latence. Cependant, les fonctions ainsi implantées sont des compositions d'opérateurs logiques élémentaires qui s'avèrent plus complexes que si l'on profite des caractéristiques des composants. Ainsi, l'addition de 183 bits est réalisée dans [LLAS13] par un transcodeur de 183 bits en entrée vers huit bits en sortie, pour coder le nombre d'entrées actives d'un fanal.

Coussy *et al.* adoptent, eux, une approche différente dans [CCWC15]. Le but est d'adapter les principes de l'algorithme de la Section 1.3.3.1 modélisant les réseaux à cliques aux unités de calculs canoniques en électronique numérique. Ainsi, la règle de comparaison WTA présentée dans la Section 1.3.3.2 est remplacée par des équations logiques. Ces dernières sont mieux adaptées aux traitements effectuées en électronique numérique. L'avantage de cette approche est d'adapter les traitements à l'utilisation de l'électronique numérique, ce qui permet au circuit d'occuper 57% de

ressources matérielles en moins que dans [JOGG12], une version antérieure du circuit de [JOGG14]. Toutefois, le circuit optimise les fonctions du modèle en termes de complexité, au risque de perdre un degré de redondance des données.

2.1.4 Approche mixte analogique/numérique proposée

L'approche que nous proposons pour concevoir un circuit implantant les fonctions du modèle des réseaux de neurones à cliques est différente de celles proposées précédemment. En effet, nous proposons d'utiliser des assemblage d'éléments analogiques ou numériques de faible complexité dont les caractéristiques se rapprochent naturellement des fonctions que nous cherchons à implanter.

Ainsi, comme les fanaux communiquent à l'aide d'informations binaires, nous proposons d'utiliser l'électronique numérique pour implanter les fonctions de communication entre fanaux. En conséquence, les données échangées par les fanaux sont naturellement résistantes au bruit environnant et il n'y a donc pas besoin de régénérer le signal d'une connexion incidente à un fanal.

En ce qui concerne les fonctions à l'intérieur d'un fanal, plutôt que d'utiliser un assemblage complexe de fonctions numériques, nous préférons utiliser des dispositifs qui s'y prêtent plus naturellement. En effet, les calculs arithmétiques simples présents dans un fanal ont des implantations peu complexes en électronique analogique. Ainsi, dans [LLAS13], nous avons également réalisé une implantation analogique des réseaux de neurones à cliques. Par exemple, l'addition de 183 signaux représentés par des courants est effectuée par un simple nœud et propagée par un miroir de courant en électronique analogique. De plus, même si les équations mises en œuvre dans [CCWC15] pour remplacer la comparaison WTA simplifient le circuit de Jarollahi *et al.*, elles restent des combinaisons d'éléments logiques dont le rôle peut être simplement rempli par des éléments analogiques. En effet, dans [LLAS13], seulement deux transistors par neurones sont nécessaires pour une intégration analogique de la comparaison WTA.

De plus, pour ces fonctions simples, il a été montré dans [LLAS13] que les dispositifs analogiques consomment 1 165 fois moins d'énergie que leur équivalent numérique. Le constat que des dispositifs analogiques consomment moins d'énergie que leur équivalent numérique est également effectué pour d'autres modèles de réseaux de neurones dans [GTBB09] et [JBTH12].

En conséquence, nous allons utiliser l'électronique analogique pour concevoir les fonctions du fanal et l'électronique numérique pour la communication entre ces derniers. Le circuit dans sa globalité est donc une implantation mixte analogique/numérique des réseaux de neurones à cliques.

Enfin, pour réaliser l'intégration matérielle d'un réseau à cliques, nous choisissons d'utiliser des fonctions à base de transistors MOS (Metal Oxide Semiconductor) sur silicium. Ces derniers permettent une réalisation compacte de fonctions analogiques et numériques. En utilisant cette technologie, plusieurs niveaux de métal sont disponibles, les interconnexions et les croisements de pistes électriques sont donc faciles à réaliser matériellement.

De plus, l'intégration matérielle grâce à la technologie CMOS est répandue dans la majorité des SoC (System-on-Chip). Il est donc possible d'intégrer un circuit implantant le modèle des réseaux de neurones à cliques en CMOS dans un SoC plutôt que sur un circuit à part, pour réduire les

coûts de fabrication du système.

Plus particulièrement, nous utilisons le kit de conception ST CMOS 65 nm fourni par ST Microelectronics. La longueur minimale du canal d'un transistor dans notre circuit est donc de 65 nm. Cette technologie a été mise en circulation en 2006, et est donc suffisamment mature. Ce kit de conception permet de réaliser des pistes métalliques sur sept niveaux de métal, nommés $M1$ à $M7$ dans la suite du document.

Nous avons ainsi effectué tous les choix technologiques, et pouvons donc commencer à nous intéresser à l'architecture de l'implantation des réseaux à cliques. Dans la prochaine section, nous proposons plusieurs organisations pour notre circuit centrées autour d'un élément de base, le cluster.

2.2 Architectures du circuit développé

Cette section a trois objectifs. Tout d'abord, nous définissons les notations relatives aux nombres d'éléments dans le réseau à cliques. Ensuite, nous proposons une structure d'intégration pour le cluster, l'unité de base au cœur des calculs dans le réseau. Enfin, nous décrivons plusieurs architectures génériques permettant d'organiser les éléments d'un réseau.

2.2.1 Calcul du nombre d'éléments constituant le réseau

Un réseau à cliques est défini en connaissant son nombre de clusters N_C, le nombre total de fanaux $N_{F_{tot}}$ et son dictionnaire, dont dépend la densité du réseau d. Pour simplifier l'étude, nous considérons que le stockage des informations dans le réseau est uniforme, c'est-à-dire que tous les fanaux sont connectés au même nombre de fanaux dans le réseau.

Le nombre de fanaux par cluster, noté N_F, est défini par :

$$N_F = \frac{N_{F_{tot}}}{N_C}. \tag{2.1}$$

Une connexion entre deux fanaux part d'un fanal émetteur et se termine par une synapse, elle-même connectée au fanal récepteur. Dans d'autres types de réseaux de neurones vus dans le Chapitre 1, chaque synapse porte un coefficient pondérant l'entrée correspondante. Dans le cas des réseaux de neurones à cliques, ce coefficient peut être vu comme binaire. S'il vaut '0', ni la connexion, ni la synapse n'existent. S'il vaut '1', la synapse existe avec un coefficient unitaire.

En considérant un stockage des messages aléatoires et uniformes, le nombre de synapses par fanal N_S, égal au nombre de connexions incidentes est donc :

$$N_S = d(N_C - 1)N_F + 1 = dN_{F_{tot}}(1 - \frac{1}{N_C}) + 1 \tag{2.2}$$

$$\text{et } \lim_{N_{F_{tot}} \to \infty} N_S = dN_{F_{tot}}.$$

N_S dépend du fait qu'un fanal peut être connecté à tous les autres fanaux des autres clusters, pondéré par la densité du réseau. De plus, une connexion par fanal est ajoutée afin de prendre en

compte une stimulation externe.

A l'échelle du réseau complet, le nombre de synapses dans le réseau $N_{S_{tot}}$ est défini par le nombre de synapses par fanal N_S multiplié par le nombre total de fanaux $N_{F_{tot}}$.

2.2.2 Structure d'un cluster

Afin de définir une ou plusieurs architectures pour circuit implantant les réseaux à cliques, nous devons identifier l'unité de calcul de base du circuit, c'est-à-dire l'unité de calcul de plus haut niveau qui, répétée plusieurs fois, forme le réseau à cliques dans sa globalité. Pour cela, nous partons de l'algorithme de la Section 1.3.3.1. La boucle concernant les itérations est mise à part car elle ne concerne pas une unité de calcul en particulier. L'unité de calcul sur laquelle le processus de fonctionnement du réseau est répété au plus haut niveau est donc le cluster. On remarque aussi cela dans l'organisation des fanaux montrée dans la Figure 1.4 de la Section 1.3.1.

Maintenant que nous avons identifié l'unité de base du circuit, nous nous intéressons à sa composition. Un cluster est composé de N_F fanaux, auxquels sont attachées N_S synapses chacun. Afin de minimiser les connexions entre ces fanaux à travers la comparaison, nous les organisons les uns à côté des autres, en deux colonnes. Nous plaçons ensuite les synapses à côté de chaque fanal auquel elles sont connectées.

La Figure 2.1 montre le schéma de l'architecture du circuit d'un cluster. Elle illustre également les paramètres nécessaires afin de calculer la taille de ce dernier, dont les définitions sont données dans le Tableau 2.1. Nous avons également fait le choix de séparer les parties analogiques et numériques des fanaux afin de limiter les interférences de la partie numérique sur la partie analogique. De plus, un bus de communication de largeur $N_{F_{tot}}$ amène l'état de tous les fanaux aux synapses pour les connecter plus facilement. Ce bus a une partie verticale et une partie horizontale de façon à ce que les clusters puissent se connecter au-dessus, au-dessous, à droite et à gauche les uns des autres. Ces connexions sont sur deux niveaux de métal différents, par exemple $M1$ pour les connexions verticales et $M2$ pour les connexions horizontales. Ces bus s'interconnectent grâce à des vias métalliques de

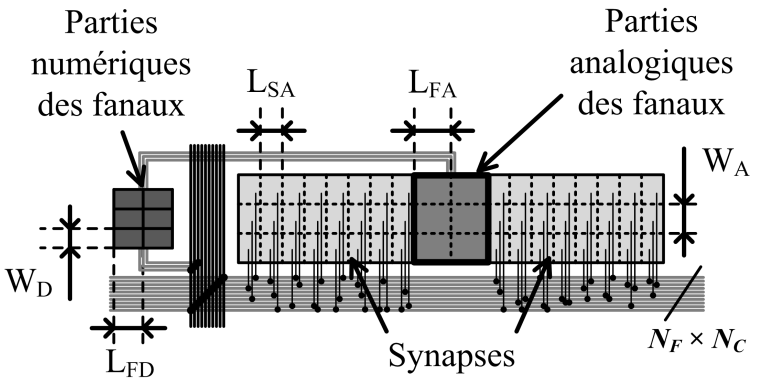

FIGURE 2.1 – Schéma d'un cluster indiquant les dimensions des éléments.

TABLEAU 2.1 – Définitions des notations utilisées pour les calculs de surface.

Notation	Paramètre correspondant
L_{FA}	Longueur d'un fanal (partie analogique)
L_{FD}	Longueur d'un fanal (partie numérique)
L_{SA}	Longueur d'une synapse
W_A	Largeur d'une synapse ou de la partie analogique d'un fanal
W_D	Largeur de la partie numérique d'un fanal
L_H	Espace entre les connexions horizontales
L_V	Espace entre les connexions verticales

la couche $M2$ vers la couche $M1$. L'espace entre les lignes verticales est désigné par L_V et celui entre les lignes horizontales par L_H.

2.2.3 Architectures envisageables du réseau

Dans les paragraphes suivants, nous nous intéressons à plusieurs façons d'organiser les clusters entre eux. Chacune de ces organisations permet d'implanter un réseau à cliques en optimisant un critère d'évaluation du circuit défini dans la Section 2.1.1, à savoir :

- la latence du circuit, en organisant les clusters en parallèle et en les interconnectant par de simples lignes métalliques ;
- et la surface de semi-conducteur, en utilisant une organisation se basant sur la réutilisation matérielle limitant ainsi le nombre de clusters physiquement réalisés.

Nous proposons ainsi trois architectures pour le circuit, allant d'une organisation complètement parallèle à une autre entièrement basée sur la réutilisation matérielle et comportant différents degrés de parallélisme concernant le nombre de clusters.

2.2.3.1 Architecture complètement parallèle

Au premier abord, nous considérons une organisation complètement parallèle où tous les clusters sont connectés entre eux. Le réseau à cliques va donc être décomposé en une matrice de clusters répartis régulièrement spatialement. Les fanaux peuvent alors échanger leurs informations de manière asynchrone dès qu'elles sont disponibles jusqu'à convergence du réseau.

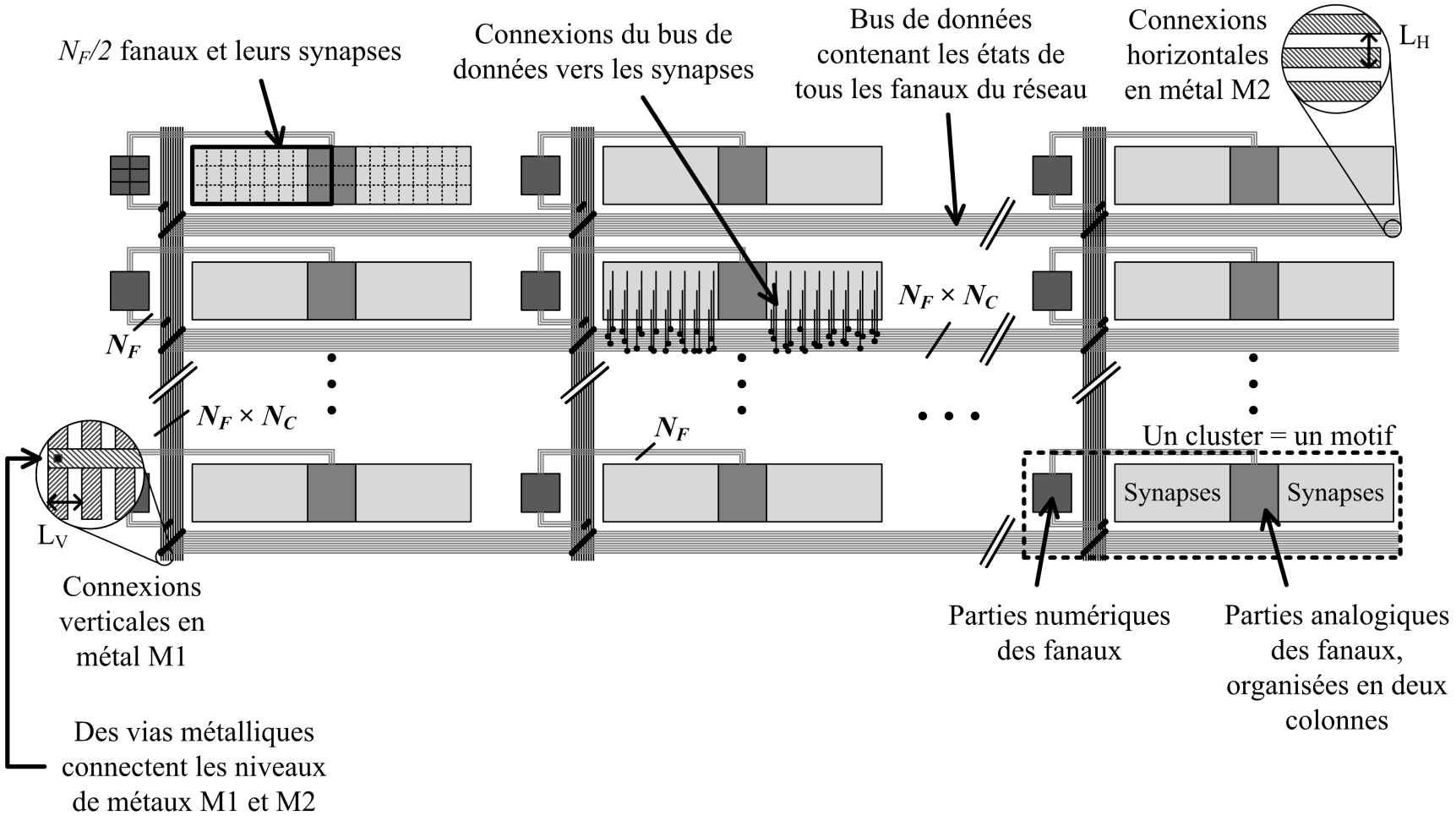

FIGURE 2.2 – Schéma de l'architecture complètement parallèle d'un réseau à cliques.

Cela permet d'utiliser un système de communication entre les fanaux composé de simples lignes métalliques, et de garder la latence du circuit au minimum. Sans dispositif de communication actif, la consommation de puissance dans les échanges d'informations entre fanaux est elle aussi minimisée. Il n'y a de plus pas de signal d'horloge cadençant la communication entre fanaux. La Figure 2.2 représente un schéma détaillé de l'organisation spatiale du réseau complet.

De cette manière, les connexions avec les sorties des fanaux sont faites sur les bus de données verticaux. Chaque synapse peut aussi être connectée à une piste d'un bus de données horizontal, comme représenté sur la Figure 2.2.

En utilisant un motif régulier, cette organisation spatiale permet une estimation aisée de la taille d'un réseau à cliques. En revanche, les fanaux peuvent ne pas recevoir les informations en même temps à cause des différentes longueurs des connexions. Cela a pour effet de fortifier les liaisons courtes, plus réactives, au détriment des plus longues. De plus, la surface de la solution de routage retenue augmente de manière quadratique avec le nombre de fanaux dans le réseau.

En outre, un réseau directement connecté par des pistes métalliques n'est pas flexible, c'est-à-dire qu'une fois fabriqué, le circuit ne peut plus être changé, et n'est pas générique. Le circuit n'est donc valable que pour une seule application.

2.2.3.2 Architectures basées sur la réutilisation matérielle

Le principe de la réutilisation matérielle, couramment utilisé en électronique numérique, peut être utilisé afin de réduire la surface de silicium occupée par le réseau. Toutefois, une architecture basée sur la réutilisation matérielle exige un ordonnancement particulier basé sur la mise en mémoire des états à chaque itération du processus de décodage. Cela impose l'utilisation d'une horloge cadençant les traitements et les échanges d'information, et a pour effet de créer des itérations dans le décodage d'un message. La durée d'une période de ce signal d'horloge est noté T_h. En revanche, l'utilisation d'une méthode de communication synchrone apporte de l'homogénéité dans les temps de propagation dans les connexions, ce qui pouvait poser problème dans une organisation complètement parallèle.

La Figure 2.3 illustre la nouvelle architecture adoptée. L'unité de calcul est composée des clusters. Elle présente un certain degré de parallélisme N_P, c'est-à-dire que N_P clusters sont effectivement présents dans cette unité et fonctionnent en parallèle. N_P peut varier de un à N_C. A l'intérieur des clusters, des synapses ont été enlevées pour profiter du caractère parcimonieux des clusters. En effet, comme un seul fanal par cluster peut être actif, $N_C - 1$ synapses peuvent être activées en même temps par fanal. Nous ne gardons donc qu'un nombre N_C de synapses par fanal, en prenant en compte la stimulation externe.

Dans le modèles des réseaux à cliques, les informations sont stockées dans la présence ou non de connexion entre deux fanaux. En supprimant les connexions directes entre les fanaux, les informations ne sont pas présentes dans le réseau. Ces informations doivent alors être stockées dans une mémoire M_C appelée mémoire des connexions, représentée dans la Figure 2.4.

Les données sont organisées par fanal tout d'abord, c'est-à-dire les indices de tous les fanaux

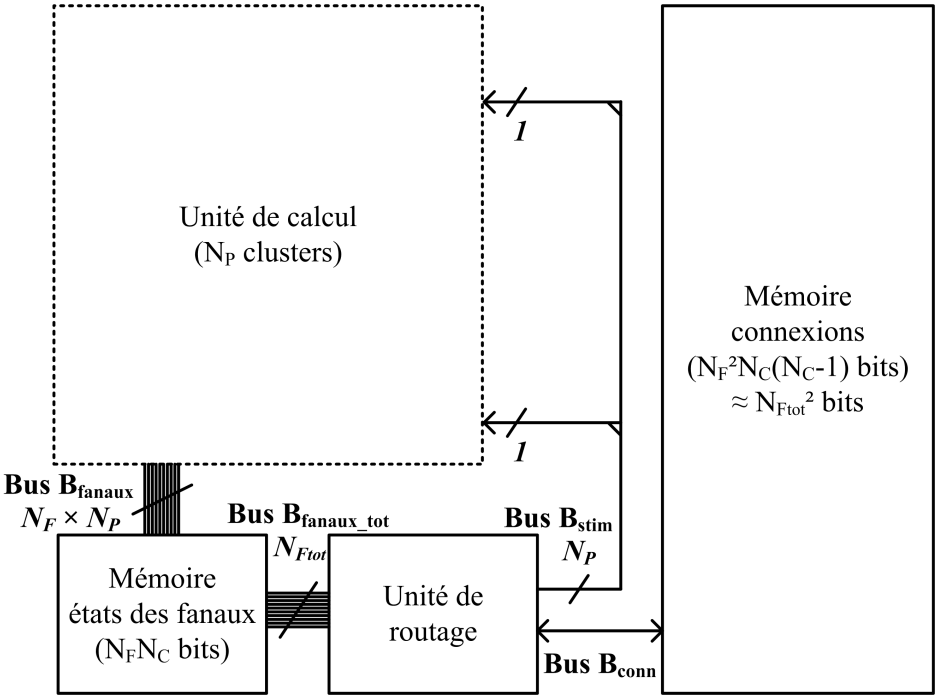

FIGURE 2.3 – Schéma de l'architecture d'un réseau de neurones à cliques basée sur la réutilisation matérielle.

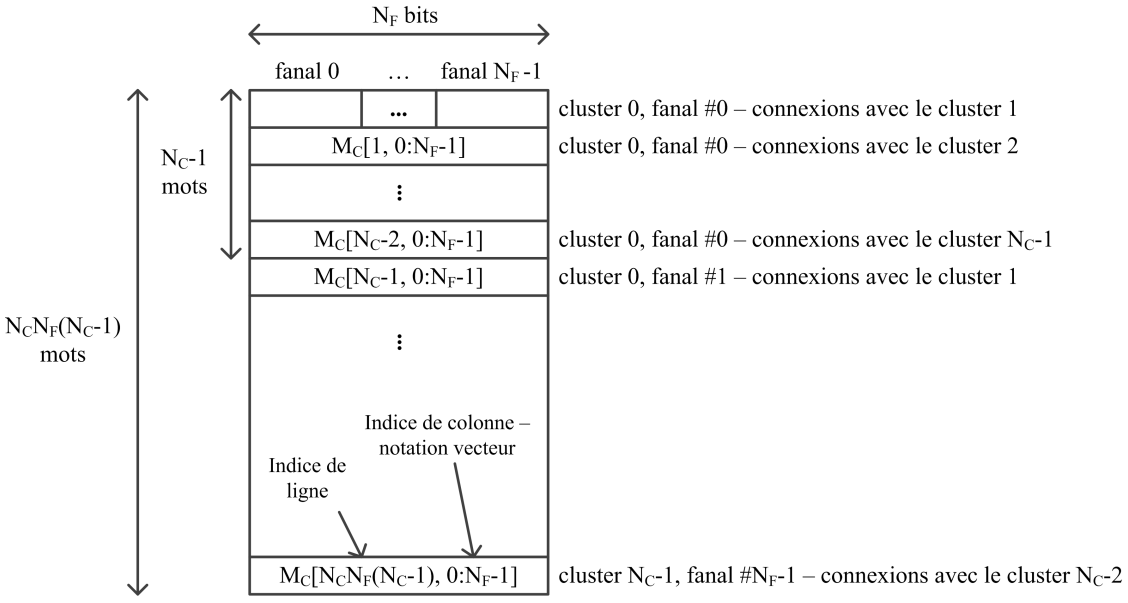

FIGURE 2.4 – Schéma de la mémoire M_C stockant les connexions entre fanaux.

auxquels un seul fanal est connecté, puis par cluster. La taille de cette mémoire M_C est incompressible et égale au minimum à $N_{S_{tot}}$, soit asymptotiquement proportionnelle à $N_{F_{tot}}^2$ sans redondance.

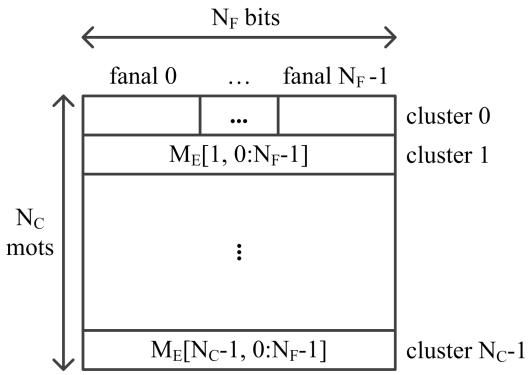

FIGURE 2.5 – Schéma de la mémoire M_E stockant les états des fanaux.

Toutes les synapses doivent de plus être présentes car toutes les connexions possibles doivent pouvoir être réalisées. Il n'est pas nécessaire d'introduire de la redondance dans cette mémoire, car les connexions entre les fanaux ont déjà un haut degré de redondance, comme expliqué dans la Section 1.3.2.

Les états des fanaux circulent dans un bus de largeur $N_{F_{tot}}$, comme dans la Section 2.2.3.1, et sont stockés dans la mémoire des états M_E, contenant $N_{F_{tot}}$ bits elle aussi. L'organisation de cette mémoire est donnée dans la Figure 2.5. Un état est représenté par un bit. Chaque ligne correspond à un cluster, et les colonnes sont les indices des différents fanaux du cluster.

Ces données mémorisées sont ensuite traitées par l'unité de routage. Son but est de générer les signaux qui vont stimuler les synapses qui ont besoin de l'être en fonction des états courants des fanaux et des connexions réalisées entre les fanaux. Ces dernières sont stockées dans la mémoire des connexions M_C, identique à celle présentée en Section 2.2.3.1.

L'unité de routage doit donc, pour chaque cluster, appliquer aux états des fanaux un masque représentant leur connexion ou non vers les fanaux du cluster considéré. Cette opération revient à effectuer des 'ET' logiques entre deux lignes des mémoires M_E et M_C. Les résultats sont ensuite destinés à être fournis au cluster correspondant. Or, comme au plus un seul fanal est actif par cluster, les N_F bits provenant du même cluster destinés à un fanal peuvent être compressés en un seul : un cluster fournit une contribution à ce fanal ou non. Cela correspond à faire un 'OU' logique sur les bits résultants de l'opération 'ET' entre deux lignes des mémoires M_E et M_C. Enfin, pour diminuer la largeur du bus partant vers les clusters, la transmission se fait en série et ces informations sont extraites au sein-même du cluster. Chaque cluster reçoit donc un vecteur de bits de taille $N_F N_C$. L'algorithme de traitement est le suivant :

Pour tout cluster $i = 0$ to $N_C - 1$
  Pour tout fanal $j = 0$ to $N_F - 1$
    Pour tout cluster $k \neq i$
      Si $i < k$ alors
        'ET' logique entre la ligne k de la mémoire M_E et la ligne correspondante dans la mémoire M_C, d'indice $iN_CN_F + (k-1)N_F + j$.
      Sinon
        'ET' logique entre la ligne k de la mémoire M_E et la ligne correspondante dans la mémoire M_C, d'indice $iN_CN_F + kN_F + j$.
      Fin
      'Ou' logique sur tous les bits du résultats et stockage du bit résultant.
    Fin
  Fin
  Sérialisation des données par cluster : S[0 :$N_F(N_C - 1) - 1$].
Fin

D'un point de vue matériel, l'unité de routage n'est donc composée que de N_F portes logiques 'ET', d'un multiplexeur implantant la condition, d'une porte logique 'OU' à N_F entrées, et d'une mémoire de taille $N_F N_C^2$ pour la sérialisation. Les boucles sont réalisées en itérant sur le même matériel, et ne coûtent donc que du temps, soit $N_F N_C(N_C - 1)$ cycles d'horloge pour le routage, plus $N_F N_C$ cycles pour la sérialisation et $N_F N_C$ cycles pour la parallélisation des données. La latence supplémentaire introduite par la réutilisation matérielle par rapport à l'organisation complètement parallèle entre deux itérations du processus de récupération de message (cf. algorithme de la Section 1.3.3.1) est donc de $N_F N_C(N_C - 1) + 2N_F N_C$ cycles d'horloge.

Un cluster est constitué conformément à la Figure 2.6. Son architecture diffère de celui de la Figure 2.1 sur deux éléments. Tout d'abord, il comporte moins de synapses au total, c'est-à-dire $N_F N_C$ synapses. Ensuite, le bus horizontal de connexions est remplacé par la liaison série provenant

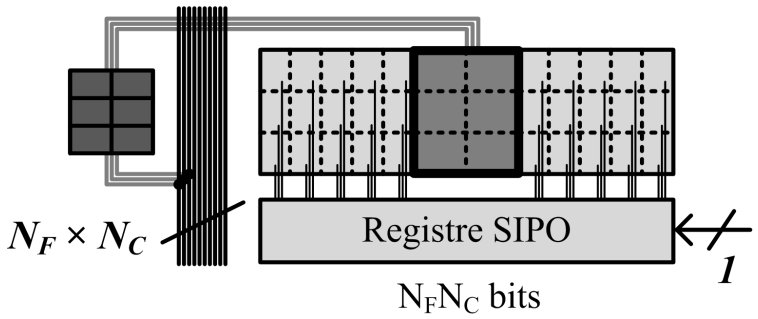

FIGURE 2.6 – Schéma d'un cluster s'interfaçant avec la méthode de communication numérique proposée.

de l'unité de routage. Un registre SIPO (Serial In - Parallel Out) parallélise le vecteur de bits et envoie chaque bit à la synapse correspondante. Sa taille est de $N_F N_C$ points mémoires. Sa complexité est donc du même ordre que celle du cluster.

Cette organisation globale peut être optimisée selon deux critères :

- le temps de réponse du réseau est minimal en implantant en parallèle tous les clusters ;
- la surface du réseau diminue si l'on n'implante qu'un cluster et que l'on itère le processus sur ce matériel, mais le temps de réponse augmente.

2.2.3.2.1 Organisation multi-clusters

La première approche consiste à implanter tous les clusters de façon à effectuer tous les traitements d'information en parallèle. Cela offre le meilleur temps de réponse pouvant être obtenu avec le système de communication synchrone.

La Figure 2.7 détaille cette organisation. Chaque cluster effectue ses opérations de comparaison et d'activation des fanaux en parallèle. Cette approche est celle qui se rapproche le plus de l'organisation décrite dans la Section 2.2.3.1.

2.2.3.2.2 Organisation mono-cluster

L'approche mono-cluster consiste à diminuer encore plus la surface du réseau en n'utilisant qu'un seul cluster sur lequel on va itérer le traitement. Cette approche augmente néanmoins le

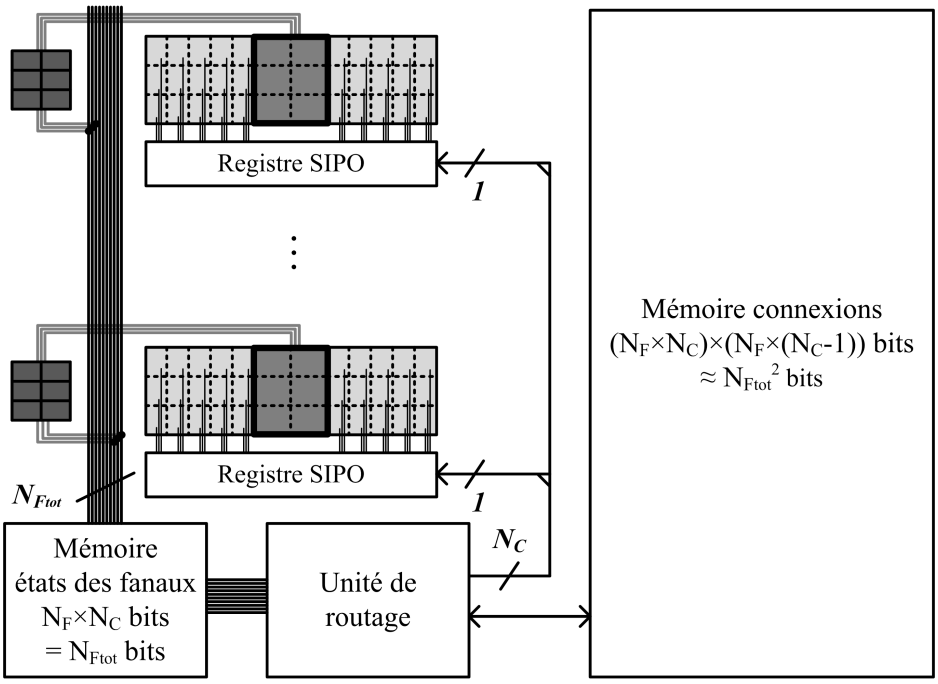

FIGURE 2.7 – Schéma de l'organisation conservant tous les clusters en parallèle.

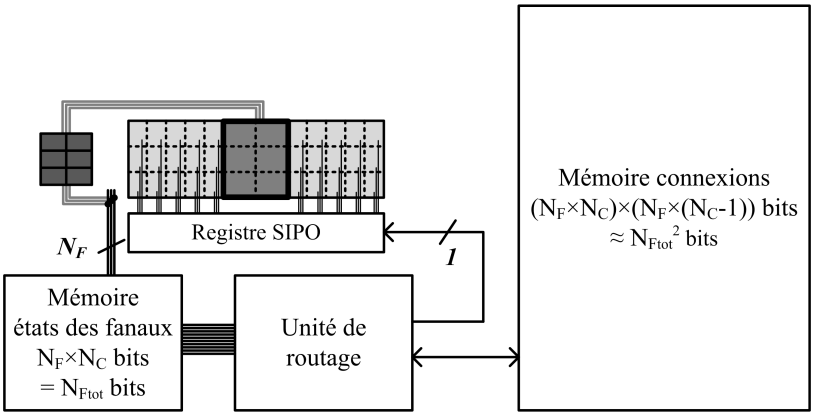

FIGURE 2.8 – Schéma de l'organisation ne conservant qu'un seul cluster pour le traitement.

temps de traitement total.

La Figure 2.8 montre un schéma de cette organisation. Un seul cluster est gardé, et met à jour progressivement la mémoire des états au cours d'une itération du décodage. La surface des fanaux et des synapses est donc divisée par N_C par rapport à celle de l'organisation multi-clusters. La surface des connexions verticales, quant à elle, est divisée par N_C^2, car la longueur et la largeur du bus sont toutes les deux divisées par N_C. La latence du traitement est en revanche augmentée d'un facteur N_C par rapport à celle de l'organisation multi-clusters, car l'opération est effectuée N_C fois par itération. L'unité de routage voit elle aussi sa surface réduite d'un facteur N_C par rapport à celle de l'organisation multi-clusters, grâce à la simplification de la sérialisation et des registres SIPO, au prix d'une latence augmentée elle aussi d'un facteur N_C par rapport à celle de l'organisation multi-clusters.

Les trois organisations proposées ayant été décrites, nous allons maintenant comparer la complexité de ces dernières pour un grand nombre de fanaux.

2.3 Comparaison des architectures

Une fois les différentes possibilités de structure spatiale du réseau à cliques définies, nous pouvons estimer leur surface quand le nombre de fanaux augmente. Pour cela, nous l'exprimons en fonction de $N_{F_{tot}}$ et en calculons la limite quand $N_{F_{tot}}$ tend vers l'infini. Cela nous donne une idée du degré de dépendance de la surface de silicium occupée en fonction du nombre de fanaux dans le réseau. Nous estimons aussi la latence du traitement dans chacune des trois organisations.

2.3.1 Calcul de l'aire du réseau

Dans tous les cas d'organisation présentés dans la section précédente, la surface d'un réseau à cliques peut être décomposée en plusieurs aires de composants élémentaires et de connexions. La surface du réseau A_{reseau} est donc égale à la somme des surfaces des fanaux (parties analogiques et

numériques) $A_{fanaux_{tot}}$, des synapses $A_{synapses_{tot}}$, des mémoires des connexions A_{mem_conn} et des états A_{mem_etats}, de l'unité de routage $A_{routage}$, et enfin de la surface des connexions horizontales et verticales, respectivement A_{h_conn} et A_{v_conn} :

$$A_{reseau} = A_{fanaux_{tot}} + A_{synapses_{tot}} + A_{mem\_conn} + A_{mem\_etats} + A_{routage} + A_{v\_conn} + A_{h\_conn}. \quad (2.3)$$

Nous devons donc exprimer chacune des surfaces composant la surface du réseau. Pour cela, nous utilisons les longueurs de chaque élément que nous avons définies dans le Tableau 2.1 de la section précédente, ainsi que le nombre de ces éléments dans le réseau.

2.3.1.1 Calcul de l'aire des fanaux

Tout d'abord, la surface occupée par les fanaux dans le réseau ne dépend que du nombre de fanaux dans le réseau et de leurs dimensions, comme le montre le Tableau 2.2 :

TABLEAU 2.2 – Expression de la surface des fanaux pour chaque organisation du réseau.

Organisation du réseau	Expression de $A_{fanaux_{tot}}$	$\lim_{N_{F_{tot}} \to \infty} A_{fanaux_{tot}}$
Organisation complètement parallèle	$N_{F_{tot}}(L_{FA}W_A + L_{FD}W_D)$	$\propto N_{F_{tot}}$
Organisation numérique multi-clusters	$N_{F_{tot}}(L_{FA}W_A + L_{FD}W_D)$	$\propto N_{F_{tot}}$
Organisation numérique mono-cluster	$\frac{N_{F_{tot}}}{N_C}(L_{FA}W_A + L_{FD}W_D)$ $= N_F(L_{FA}W_A + L_{FD}W_D)$	$\propto N_F$

Dans l'organisation complètement parallèle et l'organisation numérique multi-clusters, tous les fanaux sont implantés physiquement. Leur nombre est donc égal à $N_{F_{tot}}$, et la surface totale de fanaux est donc proportionnelle à $N_{F_{tot}}$. Dans l'organisation mono-cluster, il n'y a plus qu'un cluster implanté physiquement. Le nombre de fanaux comme leur surface sont donc divisés par N_C par rapport aux deux autres organisations.

2.3.1.2 Calcul de l'aire des synapses

De même, la surface occupée par les synapses est proportionnelle à leur nombre N_S ou N_C selon l'organisation (dont les expressions sont données dans la section précédente) et à leur taille unitaire, comme montré dans le Tableau 2.3.

On s'aperçoit que les deux organisations numériques permettent de diminuer le degré de dépendance de la surface occupée par les synapses en fonction du nombre de fanaux dans le réseau.

TABLEAU 2.3 – Expression de la surface des synapses pour chaque organisation du réseau.

Organisation du réseau	Expression de $A_{synapses_{tot}}$	$\lim_{N_{F_{tot}} \to \infty} A_{synapses_{tot}}$
Organisation complètement parallèle	$N_{F_{tot}} N_S L_{SA} W_A$ $= dN_{F_{tot}}^2 \Big((1 - \frac{1}{N_C}) + 1 \Big) L_{SA} W_A$	$\propto dN_{F_{tot}}^2$
Organisation numérique multi-clusters	$N_{F_{tot}} N_C L_{SA} W_A$	$\propto N_{F_{tot}} N_C$
Organisation numérique mono-cluster	$N_F N_C L_{SA} W_A$ $= N_{F_{tot}} L_{SA} W_A$	$\propto N_{F_{tot}}$

L'organisation mono-cluster divise de plus par N_C la surface occupée par les synapses, par rapport à l'organisation multi-clusters.

2.3.1.3 Calcul de l'aire des unités numériques

Les unités numériques n'existent que dans les organisations se basant sur la réutilisation matérielle. Ce sont les mémoires des connexions et des états, ainsi que l'unité de routage.

La mémoire des connexions est la même pour chacune des deux organisations basées sur la réutilisation matérielle. Comme décrit dans la section précédente, elle contient approximativement $N_{F_{tot}}^2$ bits. Sa surface, directement proportionnelle au nombre de bits qu'elle contient, est donc asymptotiquement proportionnelle à $N_{F_{tot}}^2$.

La mémoire des états contient les états de tous les fanaux, à raison d'un bit par fanal. Cette mémoire contient donc $N_{F_{tot}}$ bits, et sa surface est donc asymptotiquement proportionnelle à $N_{F_{tot}}$.

En ce qui concerne l'unité de routage, nous incluons dans sa surface les registres SIPO distribués sur chacun des clusters. Dans l'organisation multi-clusters, comme expliqué dans la Section 2.2.3.2, l'unité de routage est composée d'une partie combinatoire comprenant un nombre de portes logiques élémentaires proportionnel à N_F. De plus, les mémoires pour la sérialisation puis la parallélisation des résultats vers les synapses contiennent chacune $N_{F_{tot}}(N_C - 1)$ bits. La surface de l'unité de routage, somme de ces deux surfaces, est donc asymptotiquement proportionnelle à $N_{F_{tot}} N_C$ quand le nombre total de fanaux augmente. Dans l'organisation mono-cluster, le nombre de portes logiques élémentaires est le même, mais la mémoire pour la sérialisation puis parallélisation des données est divisée par N_C, car il n'y a plus que les données d'un cluster à transmettre. La surface de l'unité de routage devient donc asymptotiquement proportionnelle à $N_{F_{tot}}$ quand le nombre total de fanaux augmente.

2.3.1.4 Calcul de l'aire des connexions

La surface occupée par les connexions verticales A_{v_conn} peut être vue comme N_C fois celle occupée par les connexions verticales dans un cluster. Cette dernière dépend de la largeur du bus et de la longueur des pistes dans un cluster, soit la largeur des fanaux plus la superposition avec les connexions horizontales ou la largeur du registre SIPO W_{SIPO}. Le Tableau 2.4 montre l'expression de cette surface totale pour chaque organisation.

TABLEAU 2.4 – Expression de la surface des connexions verticales pour chaque organisation du réseau.

Organisation du réseau	Expression de A_{v_conn}	$\lim_{N_{F_{tot}} \to \infty} A_{v\_conn}$
Organisation complètement parallèle	$N_C N_F N_C L_V \left(\frac{N_F}{2} W_A + N_F N_C W_H\right)$ $= N_C N_{F_{tot}}^2 L_V \left(\frac{1}{2N_C} W_A + W_H\right)$	$\propto N_{F_{tot}}^2 N_C$
Organisation numérique multi-clusters	$N_C N_F N_C L_V \left(\frac{N_F}{2} W_A + W_{SIPO}\right)$ $= N_{F_{tot}}^2 L_V \left(\frac{W_A}{2} + \frac{W_{SIPO}}{N_F}\right)$	$\propto N_{F_{tot}}^2$
Organisation numérique mono-cluster	$N_F L_V \left(\frac{N_F}{2} W_A + W_{SIPO}\right)$ $= N_F^2 L_V \left(\frac{W_A}{2} + \frac{W_{SIPO}}{N_F}\right)$	$\propto N_F^2$

Dans les organisations basées sur la réutilisation matérielle, la longueur du bus de connexions verticales ne dépend plus de la largeur du bus de connexions horizontales. Un facteur N_C est gagné par rapport à l'organisation complètement parallèle. De plus, dans l'organisation mono-cluster, la longueur du bus vertical est N_C fois moins long que dans l'organisation multi-clusters, et est aussi N_C fois moins large. La surface des connexions verticales est donc divisée par N_C^2 dans l'organisation mono-cluster, par rapport à l'organisation multi-clusters.

On procède de la même manière pour déterminer l'aire des connexions horizontales A_{h_conn}, sans compter la surface où se superposent les connexions horizontales et verticales. Le Tableau 2.5 montre l'expression de cette surface totale pour chaque organisation.

Les connexions horizontales sont remplacées par le registre SIPO déjà comptabilisé dans l'aire de l'unité de routage pour les organisations se basant sur la réutilisation matérielle. Leur aire est donc nulle pour ces organisations.

TABLEAU 2.5 – Expression de la surface des connexions horizontales pour chaque organisation du réseau.

Organisation du réseau	Expression de A_{h_conn}	$\lim_{N_{F_{tot}} \to \infty} A_{h\_conn}$
Organisation complètement parallèle	$2N_F N_C(L_{FA} + L_{FD} + N_S L_{SA})W_H$ $= 2N_{F_{tot}} W_H(L_{FA} + L_{FD} + N_S L_{SA})$	$\propto N_{F_{tot}}^2$
Organisation numérique multi-clusters	0	0
Organisation numérique mono-cluster	0	0

2.3.1.5 Comparaison des surfaces des différentes organisations

Pour résumer cette étude, le Tableau 2.6 présente les variations asymptotiques de chaque surface en fonction du nombre total de fanaux.

TABLEAU 2.6 – Résumé des dépendances asymptotiques de chaque surface en fonction du nombre de fanaux dans le réseau.

Valeur asymptotique de la surface considérée	Organisation complètement parallèle	Organisation numérique multi-clusters	Organisation numérique mono-cluster
$\lim_{N_{F_{tot}} \to \infty} A_{fanaux_{tot}}$	$\propto N_{F_{tot}}$	$\propto N_{F_{tot}}$	$\propto N_F$
$\lim_{N_{F_{tot}} \to \infty} A_{synapses_{tot}}$	$\propto dN_{F_{tot}}^2$	$\propto N_{F_{tot}} N_C$	$\propto N_{F_{tot}}$
$\lim_{N_{F_{tot}} \to \infty} A_{mem\_conn}$	0	$\propto N_{F_{tot}}^2$	$\propto N_{F_{tot}}^2$
$\lim_{N_{F_{tot}} \to \infty} A_{mem\_etats}$	0	$\propto N_{F_{tot}}$	$\propto N_{F_{tot}}$
$\lim_{N_{F_{tot}} \to \infty} A_{routage}$	0	$\propto N_{F_{tot}} N_C$	$\propto N_{F_{tot}}$
$\lim_{N_{F_{tot}} \to \infty} A_{v\_conn}$	$\propto N_{F_{tot}}^2 N_C$	$\propto N_{F_{tot}}^2$	$\propto N_F^2$
$\lim_{N_{F_{tot}} \to \infty} A_{h\_conn}$	$\propto N_{F_{tot}}^2$	0	0

Nous pouvons maintenant estimer le degré de dépendance de l'aire totale du réseau A_{reseau} en fonction du nombre total de fanaux. Comme rappelé par l'équation (2.3), cette aire est la somme de toutes les aires estimées dans le Tableau 2.6. Dans les trois organisations proposées, la limite de la somme de toutes les aires élémentaires quand $N_{F_{tot}}$ tend vers l'infini est proportionnelle à $N_{F_{tot}}^2$.

Cette dépendance ne peut pas être réduite sans sacrifier des fonctionnalités du réseau, car pour le cas d'un réseau flexible, le degré de dépendance de la surface de la mémoire des connexions ne peut pas descendre en dessous de $N_{F_{tot}}^2$. Cependant, nous pouvons remarquer que les organisations réutilisant le matériel permettent de réduire l'augmentation de la surface de certains éléments en fonction du nombre total de fanaux. Ainsi, la surface des synapses passe d'un degré de dépendance asymptotique à $N_{F_{tot}}$ quadratique, dans l'organisation complètement parallèle, à linéaire. Les connexions horizontales sont elles supprimées. Ces surfaces en moins sont remplacées par l'ajout de la mémoire des états et de l'unité de routage. Or, leurs surfaces ne dépendent asymptotiquement que linéairement par rapport à $N_{F_{tot}}$. Ces dernières deviennent donc négligeables quand $N_{F_{tot}}$ tend vers l'infini, par rapport aux surfaces de la mémoire des connexions ou des connexions verticales. De plus, l'organisation mono-cluster permet de diviser les surfaces des fanaux, des synapses et de l'unité de routage par N_C par rapport à celles de l'organisation multi-clusters, et la surface des connexions verticales par N_C^2.

Toutefois, tous les gains de surface se font grâce au principe de réutilisation matérielle. Il y a donc un prix à payer en termes de latence et de débit de traitement pour l'utilisation de ces organisations.

2.3.2 Calcul de la latence du réseau

Nous allons maintenant estimer la latence du réseau pour toutes les organisations du réseau proposées. Cette dernière est définie dans la Section 2.1.1 comme la durée entre l'instant où sont présentés les signaux à l'entrée du circuit et l'instant où les sorties du circuit sont disponibles. Elle est estimée par le paramètre T_{CONV} qui est le temps de convergence du réseau, c'est-à-dire le temps qu'il faut au réseau pour converger vers un résultat après avoir été stimulé. Dans l'organisation asynchrone complètement parallèle décrite dans la Section 2.2.3.1, chaque cluster envoie ses informations dès qu'elles sont disponibles sans attendre le résultat des autres clusters. Il n'y a donc pas d'itération synchrone dans le processus, et le temps de réponse est plus court que son équivalent synchrone. Ainsi, dans ce cas, T_{CONV} est borné par $N_{IT}T_{cluster}$, où $T_{cluster}$ le temps de réponse d'un cluster, c'est-à-dire la durée entre la réception de données par les synapses et l'envoi du résultat par les fanaux. N_{IT} est le nombre d'itérations que le processus de décodage doit être itéré dans sa version synchrone pour arriver à un résultat stable.

En revanche, dans l'organisation multi-clusters, le temps mis par les clusters pour arriver à un résultat stable est bien de $N_{IT}T_{cluster}$. De plus, l'unité de routage rajoute de la latence entre les itérations pour calculer les données à envoyer aux synapses pour une nouvelle itération du processus de récupération des messages. Le temps pour l'unité de routage pour calculer et sérialiser les nouvelles entrées, puis pour les registres SIPO pour les amener aux synapses est de $N_F N_C(N_C - 1) + 2N_F N_C$ cycles d'horloge, comme nous l'avons décrit dans la Section 2.2.3.2. La durée d'un cycle d'horloge est T_h. Ce traitement a lieu entre deux itérations, il est donc répété $(N_{IT} - 1)$ fois.

Les traitements dans l'unité de routage de l'organisation mono-cluster sont les mêmes que ceux

pour l'organisation multi-clusters, mais doivent être répétés N_C fois, une fois pour chaque cluster dans le réseau. Le Tableau 2.7 résume les expressions de la latence que nous avons exprimée pour chaque organisation du réseau.

TABLEAU 2.7 – Expression de la latence des traitements pour chaque organisation du réseau.

Organisation du réseau	Expression de T_{CONV}
Organisation complètement parallèle	$\leq N_{IT}T_{cluster}$
Organisation numérique multi-clusters	$N_{IT}T_{cluster} + (N_{IT} - 1)(N_F N_C(N_C - 1) + 2N_F N_C)T_h$
Organisation numérique mono-cluster	$N_C N_{IT}T_{cluster} + N_C(N_{IT} - 1)(N_F N_C(N_C - 1) + 2N_F N_C)T_h$

Les organisations proposées ont donc chacune leurs avantages et inconvénients en termes de surface de silicium occupée et de latence des traitements. Cependant, c'est l'application dans laquelle le réseau est utilisée qui permettra de décider quel est le meilleur compromis entre ces organisations de réseau.

Conclusion

Après avoir estimé le nombre d'éléments nécessaires à la réalisation d'un réseau, ce chapitre a proposé trois organisations pour l'implantation circuit des réseaux à cliques. Ces organisations reposent sur un motif de base, le cluster. Le réseau peut donc être organisé de manière complètement parallèle, comme une matrice de ces éléments de base, ou en implantant seulement une partie des clusters et en itérant les traitements dessus.

Nous avons ensuite estimé la complexité d'un réseau à cliques en fonction du nombre total de fanaux, afin de voir comment augmentent sa surface et sa latence pour un grand nombre de fanaux dans le réseau. Chaque organisation a ses atouts en termes de surface de silicium occupée et de latence de traitement. Le choix de l'une des organisations est donc un compromis entre ces deux paramètres qui va dépendre de l'application dans laquelle le réseau de neurones à cliques va être utilisé.

Dans le prochain chapitre, nous nous intéressons à la conception de l'élément de base commun à toutes ces organisation : le cluster.

Chapitre 3

Conception des fonctions du cluster

Introduction

La fonctionnalité d'un réseau de neurones à cliques repose sur les fonctions de l'unité de calcul, le cluster, et la façon dont plusieurs d'entre eux sont organisés dans le réseau. Le chapitre précédent a montré une organisation basée sur un motif générique répétable à volonté, afin de construire un réseau de n'importe quelle taille. Nous avons aussi proposé d'autres méthodes de communication entre les clusters qui simplifient les connexions. Ce chapitre décrit les éléments qui composent les clusters, de façon à ce que ces derniers s'intègrent dans l'architecture de réseau proposée précédemment. Leur comportement est simulé en utilisant le kit de conception ST CMOS 65 nm, avec le simulateur *Spectre*®. Nous nous appuierons tout d'abord sur la structure du fanal décrite dans la Figure 1.5 du Chapitre 1 pour donner la structure du circuit. Puis, nous décrirons chacun des circuits réalisant les fonctions composant un fanal. Nous donnerons également les limites de ces circuits, notamment en terme de nombre d'éléments en parallèle. Enfin, nous décrirons le circuit d'un fanal dans son intégralité, pour aller vers la formation de clusters.

3.1 Choix de conception

Comme présenté dans le Chapitre 1, il existe de nombreux moyens d'implanter des réseaux de neurones en circuit, en électronique analogique, comme dans [MH03] ou [MM14], numérique dans [PPG+13], ou encore mixte dans [AF96]. Dans cette section, nous expliquons nos choix de conception en prenant en compte la composition d'un fanal, ainsi que les organisations des fanaux présentées dans le Chapitre 2, Section 2.2.3. Pour réaliser cette implantation mixte, nous avons vu dans la Section 2.1 que nous utilisons le kit de conception ST CMOS 65 nm. Les valeurs numériques issues de simulations ne sont donc valables que pour cette technologie. La tension d'alimentation choisie pour le circuit est de 1 V. Cette tension est en effet suffisamment basse pour réaliser un circuit faible consommation énergétique, et suffisamment importante pour assurer la saturation de tous les transistors en série et donc assurer leur fonctionnement nominal.

3.1.1 Utilisation du mode courant

À l'intérieur d'un fanal, les fonctions, rappelées dans la Figure 3.1, sont réalisées de manière analogique.

Cependant, la complexité de ces fonctions est impactée par le mode d'intégration : mode courant ou mode tension. Cela signifie que les traitements à l'intérieur du fanal sont effectués respectivement sur des courants ou des tensions.

Si l'on considère les fonctions du fanal présentées sur la Figure 3.1, le mode courant permet l'utilisation de circuits plus simples. Par exemple, l'addition de courants est réalisée par un simple nœud. Ainsi, plusieurs circuits intégrant des réseaux de neurones utilisent le mode courant, comme par exemple [GYMD12], [LLAS13] et [ABP+91].

En revanche, la transmission des informations d'un fanal à un autre est plus aisée si on la réalise à l'aide de tensions. Ces dernières peuvent être propagées via de simples lignes métalliques d'un bout à l'autre du circuit.

La Figure 3.2 montre la répartition des différents modes d'intégration utilisé dans un fanal. Les entrées du fanal sont portées par des tensions, puis converties en courant avant de réaliser l'addition. Cette dernière, ainsi que la comparaison sont effectuées sur des courants, puis le résultat

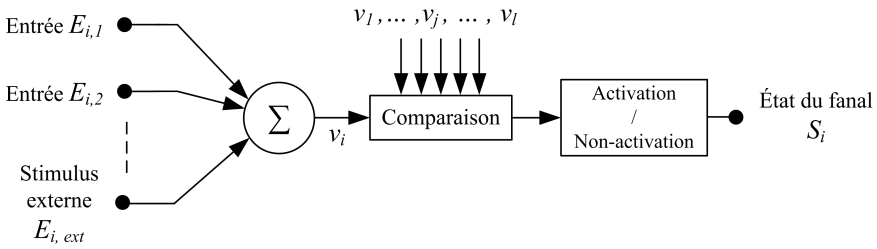

FIGURE 3.1 – Rappel des fonctions composant un fanal.

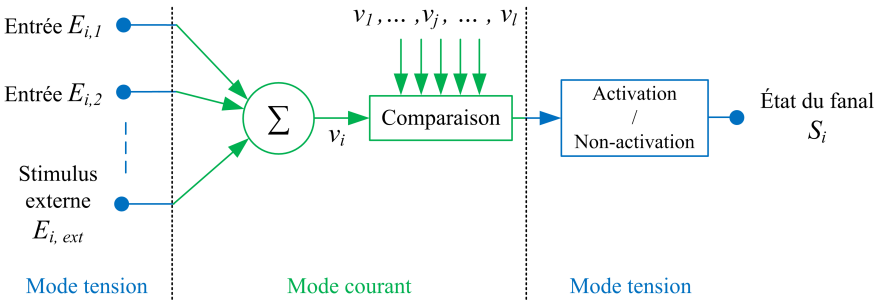

FIGURE 3.2 – Répartition des modes d'intégration des fonctions composant un fanal. Les entrées du fanal sont portées par des tensions, puis converties en courant avant de réaliser l'addition, et reconverties en tension après la comparaison.

de la comparaison est donné en tension afin de faciliter la décision sur l'activation. L'activation ou non du fanal et la propagation vers les autres fanaux se fait donc grâce à des tensions.

3.1.2 Structure du circuit au niveau d'un cluster

Dans le schéma d'un fanal de la Figure 3.1, la fonction de comparaison se fait entre tous les résultats d'addition des fanaux d'un cluster. Or, cette opération est identique dans chaque fanal, et peut être mutualisée au sein du cluster afin de faire un seul classement du nombre d'entrées actives. La structure du circuit peut alors être présentée au niveau du cluster, dans la Figure 3.3.

Chaque fanal effectue une opération d'addition des contributions en parallèle. Puis, les résultats sont comparés grâce à un module de comparaison au niveau du cluster. Enfin, chaque fanal intègre

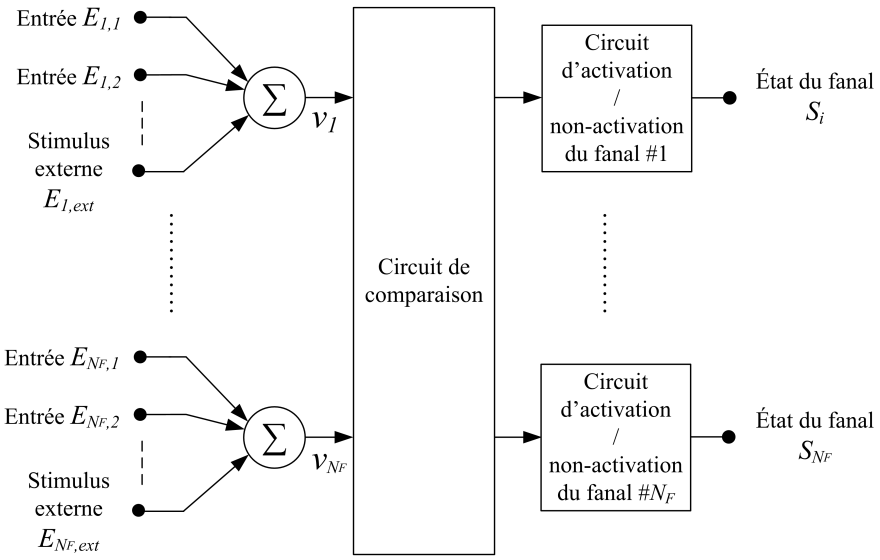

FIGURE 3.3 – Schéma de la structure du circuit d'un cluster.

une fonction de décision sur son activation ou non, qui s'effectue en parallèle avec celle des autres fanaux.

Une fois nos choix sur la conception du circuit finalisés, nous allons présenter les différents éléments du circuit d'un fanal, puis d'un cluster, plus en détail.

3.2 Addition des contributions

Un fanal reçoit des informations provenant de fanaux appartenant à d'autres clusters, ainsi qu'un signal de stimulation externe au réseau. Ces informations sont numériques, et, comme décrit en introduction de ce chapitre, sont portées par des tensions. Comme les calculs à l'intérieur du fanal sont effectués sur des courants, il est nécessaire de convertir en amont les données binaires en tension en données binaires en courant.

L'opération d'addition des signaux d'entrée peut ensuite être effectuée sur les courants binaires résultants.

3.2.1 Conversion tension-courant

La conversion tension-courant est effectuée à la terminaison de chaque connexion vers le fanal. Comme présenté dans la Section 2.2.3, chaque connexion débouche sur une synapse. Cette dernière est l'interface entre une ligne de transmission, connectant deux fanaux entre eux, et le cœur de calcul du fanal. Chaque synapse doit donc intégrer une opération de conversion d'une tension binaire vers un courant binaire.

La conversion tension-courant est réalisée grâce à une source de courant commutée, comme montré sur la Figure 3.4. Celle-ci est composée de deux transistors, M_{11} et M_{12}, plus un transistor M_5 commun à plusieurs synapses. Les transistors M_5 et M_{11} forment un miroir de courant qui recopie

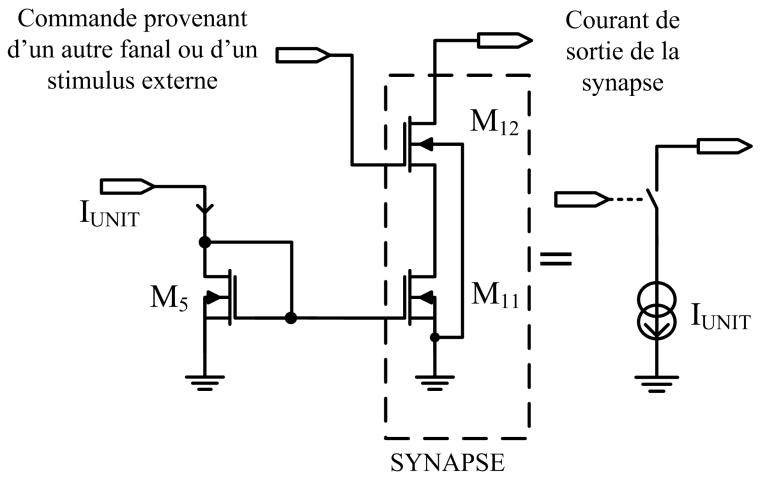

FIGURE 3.4 – Schéma électrique d'une synapse.

un courant unitaire I_{UNIT} dans la synapse. Ce courant est alors coupé ou non par le transistor M_{12}, qui agit comme un interrupteur directement commandé par la tension de la connexion venant d'un autre fanal, ou de l'extérieur. Le courant qui résulte de la conversion est alors directement envoyé vers l'opérateur d'addition. Ce courant est bien binaire, puisqu'il ne prend que deux valeurs, 0 A ou I_{UNIT}, correspondant respectivement à un '0' ou à un '1' logique, comme résumé dans le Tableau 3.1. Lorsque le courant en sortie de la synapse n'est pas nul, on dit que la synapse est *activée*.

TABLEAU 3.1 – Valeurs du courant de sortie d'une synapse en fonction de l'état logique de la connexion correspondante.

État logique de la commande	État de l'interrupteur M_{12}	Valeur du courant de sortie de la synapse
'0'	ouvert	0 A
'1'	fermé	I_{UNIT}

3.2.2 Addition en mode courant

En mode courant, l'addition est très simple à implanter : elle n'est constituée que d'un simple nœud. Comme représenté sur la Figure 3.5, dans un fanal, les synapses sont connectées en parallèle au nœud A. C'est à ce point que les courants provenant de chacune des synapses sont additionnés. Le courant résultant est ensuite recopié grâce à un miroir de courant composé de M_3 et de M_4 pour être envoyé dans le circuit de comparaison.

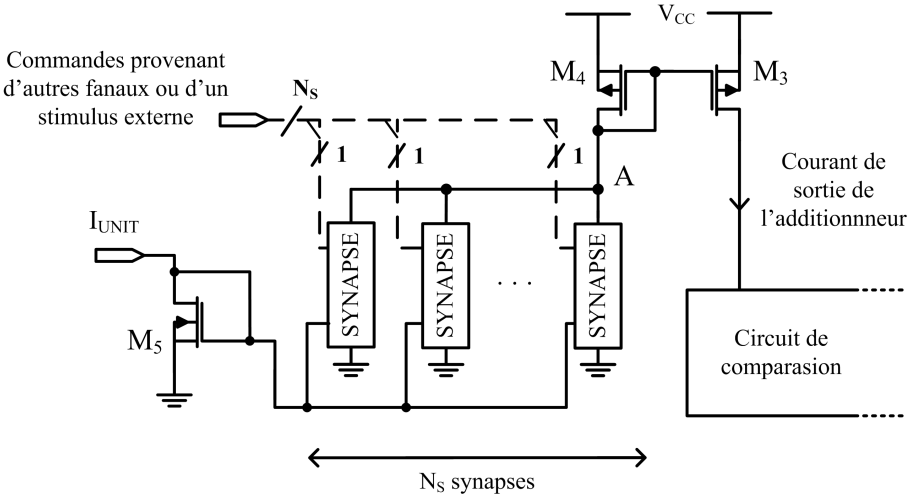

FIGURE 3.5 – Schéma électrique d'un additionneur, accompagné d'un banc de synapses.

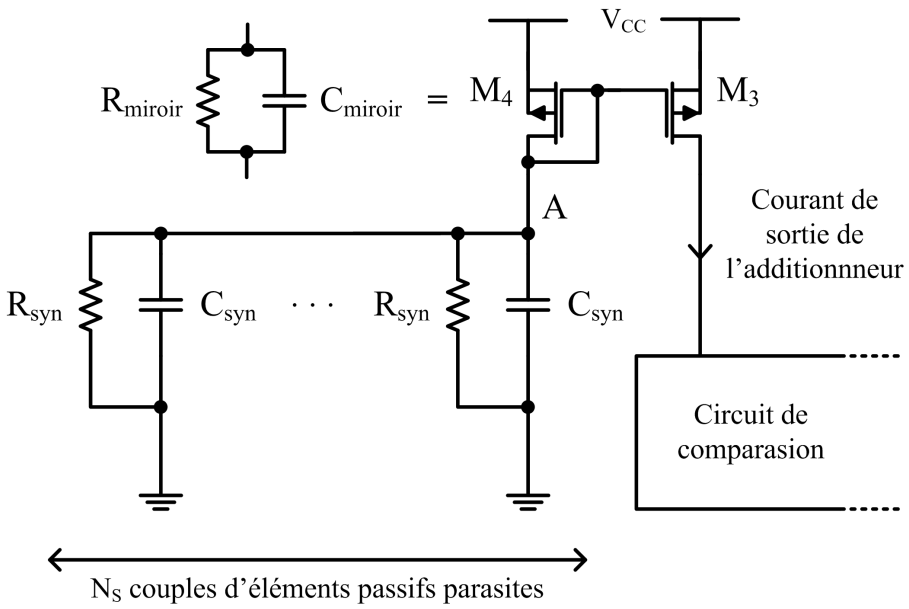

FIGURE 3.6 – Schéma électrique des éléments parasites dans les synapses, vus de l'additionneur.

3.2.3 Limitations

La grande simplicité d'une synapse peut malgré tout induire des problèmes. Certains montrent un manque de fonctionnalité au sein d'un réseau, tandis que d'autres impliquent des limitations dans l'utilisation, notamment au niveau du nombre de synapses pouvant être utilisées en parallèle.

3.2.3.1 Simulation temporelle de l'additionneur

Dans les architectures du réseau à cliques proposée dans le Chapitre 2, Section 2.2.3, chaque connexion à un fanal se traduit par la présence d'une synapse connectée à ce fanal au point A. Or, chaque synapse peut être assimilée à un dipôle RC, composé d'une résistance R_{syn} et d'une capacité C_{syn} en parallèle, comme représenté sur la Figure 3.6. La résistance R_{syn} représente la somme des résistances de l'interrupteur M_{12} et de la source de courant M_{11}. La capacité C_{syn} représente quant à elle la capacité équivalente aux capacités parasites des transistors M_{11} et M_{12}. Ces éléments passifs peuvent avoir chacun deux valeurs :

- R_{syn_on} et C_{syn_on} si l'interrupteur M_{12} est fermé ;
- R_{syn_off} et C_{syn_off} si l'interrupteur M_{12} est ouvert.

Le miroir de courant formé par M_3 et M_4 a lui aussi ses éléments passifs parasites R_{miroir} et C_{miroir}. Les valeurs de ces éléments passifs sont déterminées par simulation avec *Spectre*®, et sont données dans le Tableau 3.2.

En fonctionnement, au moins une des synapses est active. Dès qu'il y a au moins une synapse inactive en parallèle, plus le nombre de synapses en parallèle augmente, plus la capacité équivalente à toutes les synapses va augmenter.

TABLEAU 3.2 – Valeurs des éléments passifs parasites présents dans les synapses.

Élément passif	Valeur numérique simulée
R_{syn_on}	18,1 MΩ
R_{syn_off}	4 GΩ
C_{syn_on}	0,00076 fF
C_{syn_off}	0,093 fF
R_{miroir}	9,3 MΩ
C_{miroir}	0,14 fF

Le temps de réaction de l'additionneur $T_{addition}$ correspond à la durée que met l'additionneur pour avoir 90% du courant résultant au point A. La Figure 3.7 montre la valeur du temps de réaction de l'additionneur avec une tension d'alimentation V_{CC} de 1 V et un courant I_{UNIT} de 300 nA, quand une seule synapse est activée. Or, $T_{addition}$ dépend directement de la constante de temps du dipôle RC équivalent à toutes les synapses et au miroir. Cette constante de temps augmente de manière sous-linéaire avec le nombre de synapses connectées au fanal, donc avec la densité même du réseau. En effet, même si la capacité équivalente augmente de manière affine, l'ajout d'une résistance en parallèle fait diminuer la résistance équivalente, et influe sur la constante de temps. L'ordonnée à l'origine correspond à la constante de temps du miroir.

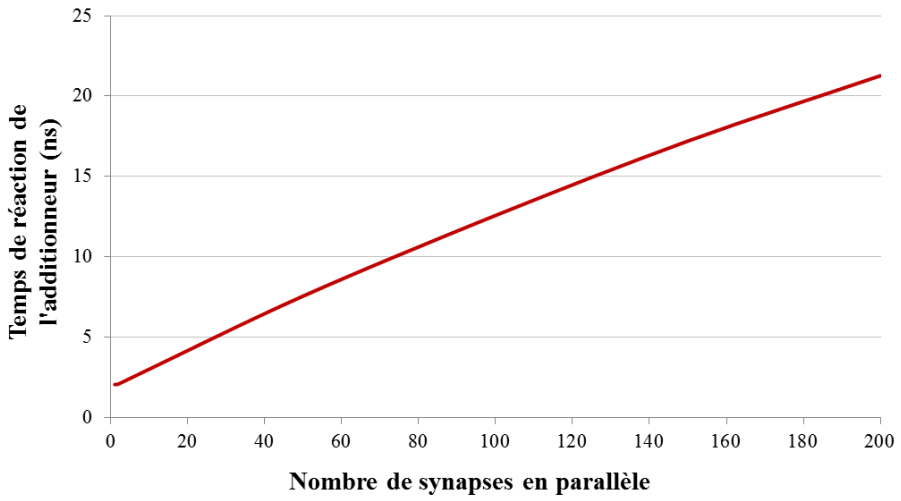

FIGURE 3.7 – Temps de réaction de l'additionneur en fonction du nombre de synapses connectées au nœud A. Une seule synapse est activée.

Une quantité plus importante d'information stockée dans le réseau à cliques se payera donc non seulement en termes de précision dans le décodage (cf. Chapitre 1), mais aussi en termes de temps de réponse des fanaux.

3.2.3.2 Perméabilité en courant de l'interrupteur

Dans la synapse, le courant I_{UNIT} est coupé ou non par un MOS, M_{12}, qui sert d'interrupteur. Or, en position ouverte, cet interrupteur laisse passer un courant résiduel estimé à 70 pA grâce à des simulations avec *Spectre*®. Lorsque plusieurs synapses sont connectées en parallèle, ce courant résiduel circule dans chacune d'elles, et tous ces courants vont aller s'ajouter au point A. Le courant résultant peut alors être suffisant pour introduire un biais dans le circuit de comparaison, et ainsi fausser la décision. En fonction du circuit de comparaison, une limite devra être fixée sur le nombre de synapses pouvant être connectées en parallèle, de façon à limiter ce courant au maximum.

3.2.3.3 Flexibilité des connexions

Enfin, les synapses telles qu'elles sont décrites dans cette section ne permettent pas un stockage d'information flexible. En effet, les connexions sont directement réalisées entre les fanaux, sans moyen de les modifier une fois le circuit mis sur puce. Pour pallier cette limitation, on peut ajouter de la programmabilité sur les connexions, en ajoutant deux transistors par connexion, M_{13} et M_{14}, Figure 3.8. Ainsi, la gestion du stockage est déportée dans une mémoire dont le contenu peut être modifié.

Cette fonctionnalité peut donc être ajoutée, au prix d'une mémoire globale de taille $N_{S_{tot}}$, et de la réalisation de toutes les connexions possibles entre les fanaux. On perd ainsi en partie l'avantage

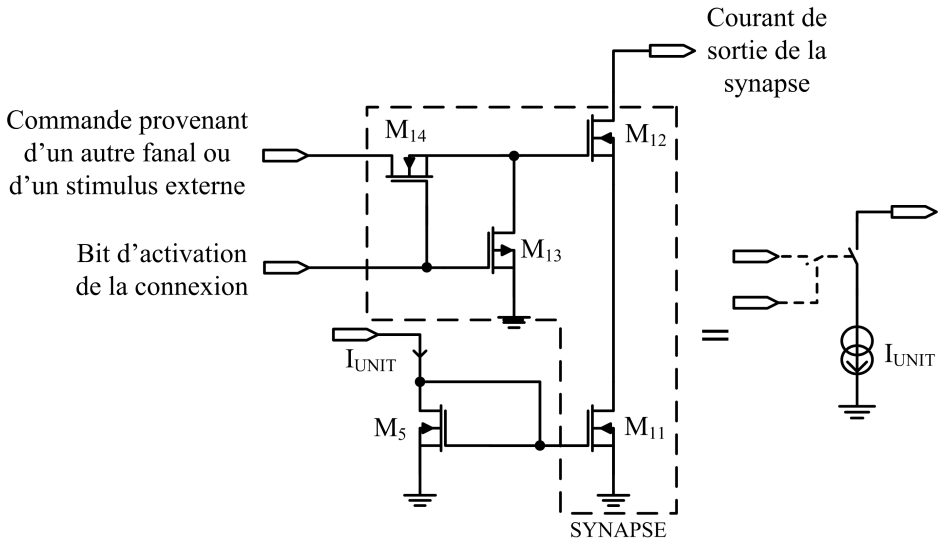

FIGURE 3.8 – Schéma électrique d'une synapse avec connexion programmable.

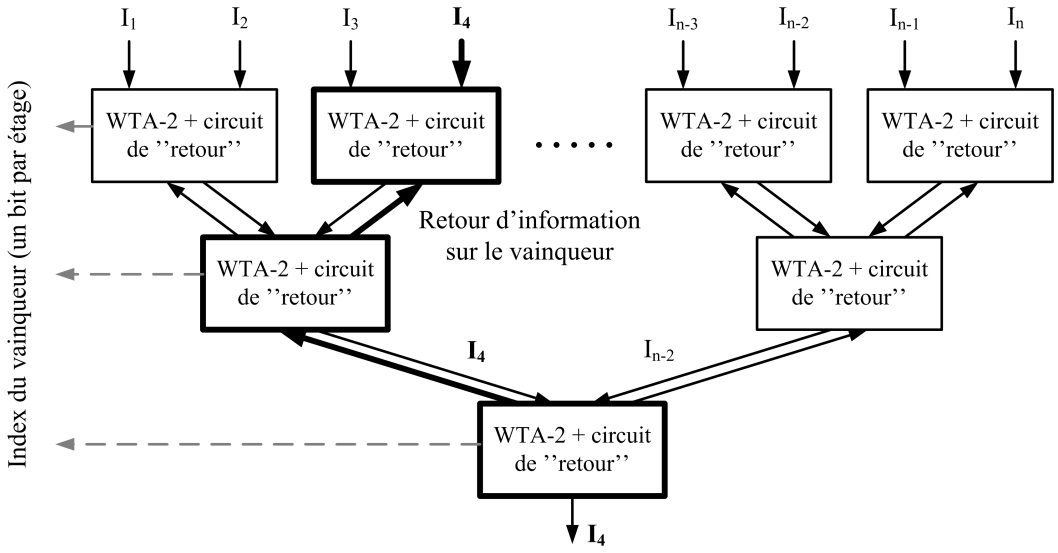

FIGURE 3.9 – Schéma bloc d'un circuit de comparaison "en arbre".

des réseaux à cliques en termes de nombre de connexions.

3.3 Comparaison du nombre de contributions

Le circuit de comparaison met en relation les courants résultant des additionneurs des fanaux à l'intérieur d'un cluster. Il utilise une (ou plusieurs) des règles de décodage présentées dans le Chapitre 1. Le but de cette section est de présenter des circuits intégrant chaque règle de décodage. La suite de l'étude utilise cependant un circuit intégrant la règle de décodage "Winner-Takes-All", car cette règle de décodage simplifie l'opération de comparaison dans le cas de messages non parcimonieux comme décrit dans la Section 1.3.2.

3.3.1 Structure de la comparaison

Le circuit de comparaison au sein d'un cluster peut être organisé en suivant deux stratégies différentes, toutes deux ayant une complexité en N_F. Tout d'abord, il est possible de faire des comparaisons de courant deux à deux à plusieurs étages, en organisant les comparaisons à la manière d'un arbre, comme montré en Figure 3.9 [DST98]. Chaque bloc de comparaison élémentaire compare deux entrées et propage la plus élevée dans l'étage supérieur. Un circuit supplémentaire est nécessaire afin de propager dans l'autre sens l'identité du plus fort.

L'autre stratégie possible est de réaliser les comparaisons en parallèle, comme sur la Figure 3.10, en jouant sur les régimes de fonctionnement des transistors [GC12, UN95, RH14]. Il est ainsi possible, en fournissant les courants d'entrée à des transistors MOS, d'étudier les points de fonctionnement de chaque transistor et de saturer uniquement les transistors recevant le plus de courant.

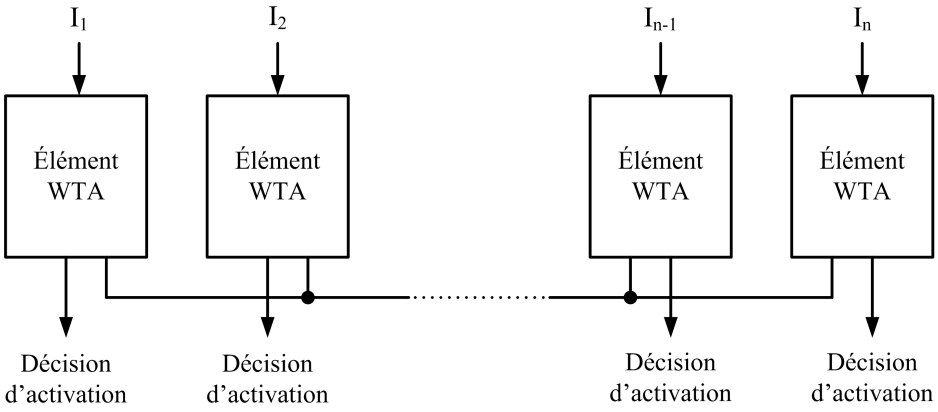

FIGURE 3.10 – Schéma bloc d'un circuit de comparaison "parallèle".

La deuxième stratégie est celle qui offre le plus de garanties en termes de généricité du fanal. En effet, elle permet d'intégrer un élément identique dans chaque fanal, alors que la première stratégie utilise des blocs élémentaires concernant deux fanaux à chaque fois. Ainsi, si la topologie du réseau venait à changer et qu'un fanal doive être ajouté à une unité de comparaison, un seul élément peut être ajouté en parallèle si l'on suit la seconde stratégie. Pour la stratégie en arbre, il faut rajouter tout un pan de l'arbre de comparaison pour ajouter un élément.

De plus, l'information sur le plus fort courant peut être lue directement dans chaque élément de comparaison, dans la deuxième stratégie. Il n'y a pas de circuit de "retour" pour connaître les fanaux vainqueurs, et cela diminue la complexité du circuit.

Pour ces dernières raisons, on choisira donc la seconde stratégie pour l'implantation de la règle de décision.

3.3.2 Opération "Winner-Takes-All"

L'opération WTA consiste à activer le fanal qui reçoit le plus de contributions dans un cluster. C'est donc une règle de décodage locale. En cas d'égalité au sein d'un cluster, les deux fanaux sont activés.

3.3.2.1 Implantation circuit de la règle "Winner-Takes-All"

La règle WTA est une règle d'activation classique dans les modèles de réseaux de neurones artificiels [LRMM88]. Ainsi, plusieurs circuits ont été développés afin d'implanter cette règle. Cependant, compte-tenu des contraintes précédemment énoncées dans cette section, c'est-à-dire structure parallèle et fonctionnement en mode courant, deux circuits sont considérés.

Mead *et al.* intègrent la règle WTA en respectant ces contraintes dans [LRMM88]. Le schéma électrique de ce circuit est donné dans la Figure 3.11. Deux transistors M_1^i et M_2^i sont intégrés dans chaque fanal, et le circuit WTA complet est formé en connectant les grilles des transistors M_1^i entre

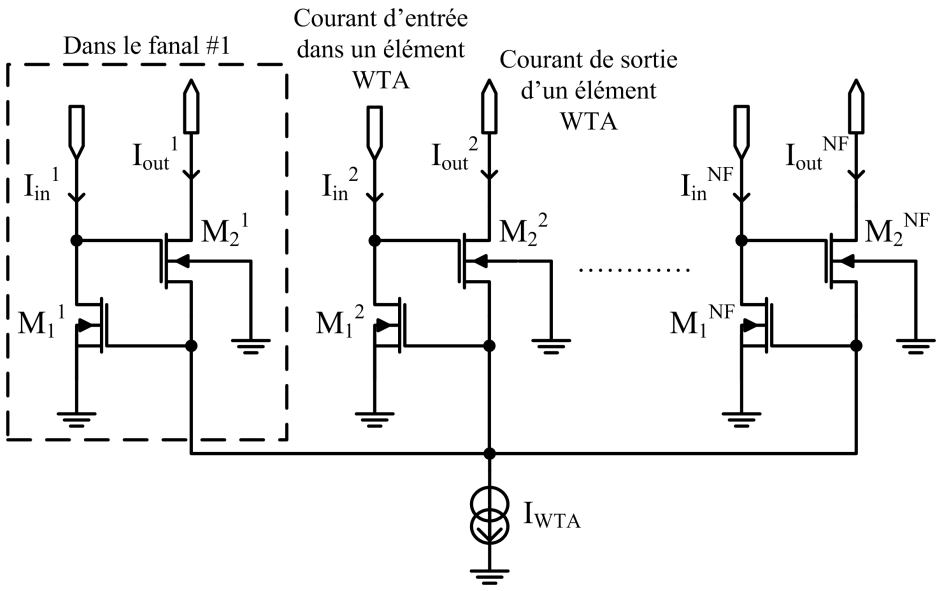

FIGURE 3.11 – Schéma électrique du circuit WTA de [LRMM88].

elles. Ce nœud commun est aussi connecté à une source de courant qui délivre un courant I_{WTA} dans la branche gagnante. Lorsqu'un courant circule dans un fanal, par exemple le fanal #1, le transistor M_1^1 permet au transistor M_2^1 de laisser passer le courant I_{WTA} dans la branche du fanal #1. Dans les autres fanaux, aucun courant ne circule.

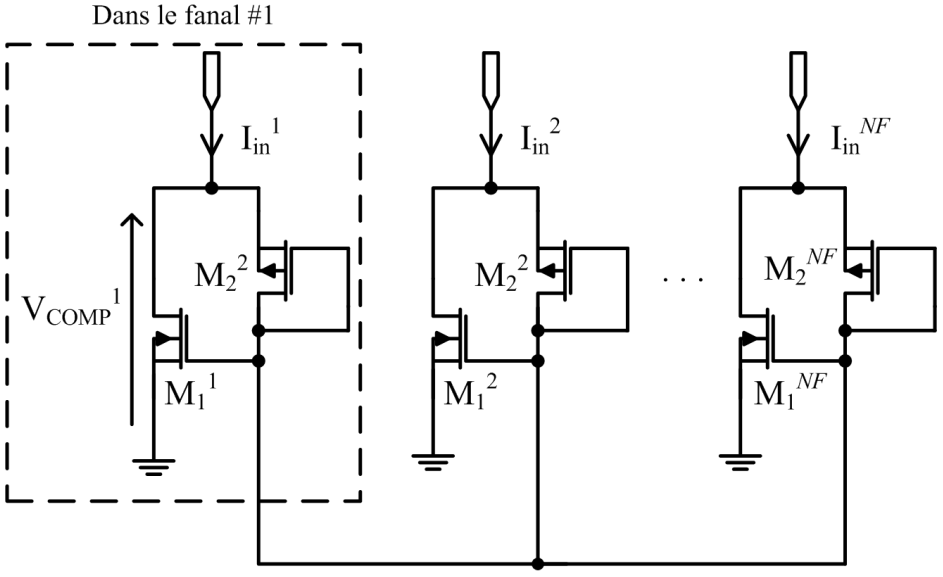

FIGURE 3.12 – Schéma électrique du circuit WTA adapté de [GC12].

Le second circuit implantant la règle WTA est adapté d'un circuit proposé par Chakrabartty

TABLEAU 3.3 – Comparaison de la complexité des circuits WTA.

Paramètre	Circuit issu de [LRMM88]	Circuit adapté de [GC12]
Nombre de transistors	2 par fanal + source de courant commune	2 par fanal
Grandeur de sortie	Courant I_{WTA}	Tension V_{COMP}

et al., utilisé dans les décodeurs de canal [GC12]. La Figure 3.12 donne le schéma électrique de ce circuit adapté. Dans [GC12], ce circuit est utilisé avec une source de courant connectée aux grilles des transistors M_1 pour remplir une fonction d'approximation d'opérations logarithme ou exponentielle. Si ce courant est nul, le circuit devient un circuit WTA. Dans ce circuit également, deux transistors M_1^i et M_2^i sont intégrés dans chaque fanal. Le circuit complet est aussi formé en reliant les grilles des transistors M_1^i entre elles. Lorsqu'un courant circule dans le fanal #1, le transistor ${M_1}^1$ devient saturé et la diode M_2^1 devient passante. Il est alors possible de lire le résultat de la comparaison aux bornes de M_1^1.

Le Tableau 3.3 récapitule les caractéristiques de chacun des circuits, en termes de nombre de transistors et de mode de sortie. Le circuit adapté de [GC12] demande moins de transistors et présente aussi l'avantage de faire la conversion courant-tension inverse des synapses. C'est donc ce circuit qui est retenu dans la suite de l'étude.

La fonctionnalité du circuit WTA choisi est vérifiée en simulation, Figure 3.13. Dans un circuit comprenant six éléments, on fixe le courant entrant dans l'élément #1 à 300 nA, et on fait varier le courant entrant de l'élément #2 de 0 à 600 nA. La Figure 3.13 montre trois phases. Si la différence entre les deux courants entrants est supérieure ou égale à 20 nA, le plus fort l'emporte. Le seuil d'activation peut être placé entre 100 mV et 400 mV, par exemple à 360 mV. Sinon, les deux courants sont considérés comme à égalité et les fanaux correspondants seront actifs tous les deux.

3.3.2.2 Limitations

Le circuit WTA a besoin d'un courant minimum en entrée pour fonctionner. Il s'agit du courant minimum que le transistor M_1 d'un élément du WTA peut laisser passer s'il n'est pas en régime bloqué. La Figure 3.14 montre l'état d'un élément WTA en fonction de la valeur du courant entrant, les courants dans les autres éléments étant nuls. Si on considère qu'un fanal est activé à partir d'une valeur de V_{COMP} de 360 mV, l'activation d'un fanal ne peut être détectée que lorsqu'un courant supérieur à 50 nA arrive dans le circuit WTA.

Le fait d'avoir un minimum de courant à atteindre pour faire fonctionner le circuit WTA peut être considéré comme une limite. Cela impose un minimum sur la valeur de I_{UNIT} et sur les courants circulant dans le circuit, et donc augmente la puissance utilisée par le circuit. Mais cela peut aussi être positif, car, comme vu en Section 3.2.3.2, cela permet d'augmenter le nombre de

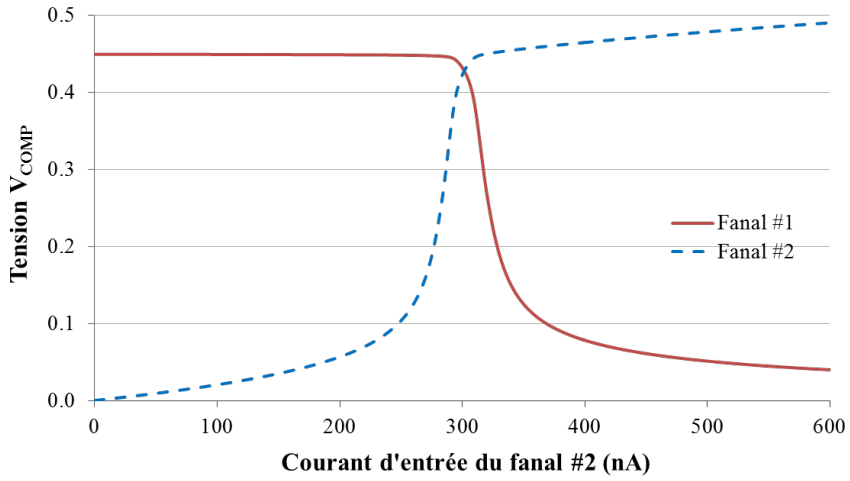

FIGURE 3.13 – Réponse du circuit WTA lors de la variation de l'un des courants entrants.

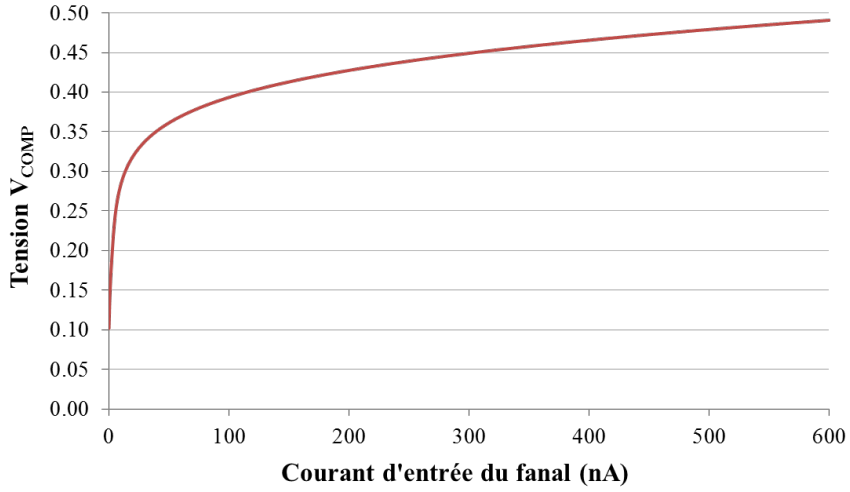

FIGURE 3.14 – Réponse d'un élément du circuit WTA lors de la variation du courant entrant de 0 à 600nA.

synapses pouvant être mises en parallèle.

De plus, le nombre d'éléments WTA pouvant être connectés en parallèle, donc le nombre de fanaux dans un cluster, est limité. En effet, dans chaque élément du circuit WTA inactif, un courant résiduel, dont la valeur est donnée à 3,57 nA en simulation, circule dans la diode bloquée. Ce courant ne peut provenir que de la branche active du circuit WTA, et est donc "prélevé" du courant passant dans M_1. Pour une valeur de courant minimale entrant dans le circuit WTA, c'est-à-dire I_{UNIT} ou 300 nA, il ne faut donc pas qu'il y ait plus de 70 éléments WTA en parallèle afin de pouvoir dépasser les 50 nA nécessaires afin d'activer l'élément WTA correspondant.

Enfin, de la même manière que dans la Section 3.2.3.1, le temps de réaction du circuit WTA

augmente avec le nombre d'éléments WTA en parallèle. Chaque élément en plus rajoute une capacité parasite en parallèle du circuit. Ainsi, le temps de réaction du circuit WTA augmente linéairement avec le nombre d'éléments WTA en parallèle, de 120 ns par élément WTA.

3.3.3 Opération "k-Winners-Take-All"

L'opération "k-Winners-Take-All" ressemble à l'opération "Winner-Takes-All", mais cette fois k fanaux sont activés. Cette opération est également réalisée dans d'autres modèles de réseaux de neurones artificiels [UN95, RH14]. Dans les réseaux de neurones à cliques, cette règle de décodage est utilisée comme règle globale, en fixant la valeur de k comme le nombre de nœuds dans une clique.

Un circuit implantant la règle k-WsTA a déjà été proposé par Hasler *et al.* [RH14]. La Figure 3.15 montre le schéma électrique de ce circuit. Il est dérivé du circuit proposé dans [LRMM88] et dont le fonctionnement est présenté dans la Section 3.3.2.1. Dans le circuit de la Figure 3.15, les sorties en courant sont limitées par un courant $I_{OUT_{WTA}}$, dont le courant I_{WTA} est un multiple, de facteur k. Dans ce cas, le courant I_{WTA} se répartit entre les k branches gagnantes.

La Figure 3.16 montre le fonctionnement d'un tel circuit. Dans un circuit de six éléments, on choisit k=2, et on fixe donc $I_{OUT_{WTA}}$ à 100 nA et I_{WTA} à 200 nA. Au début de la simulation, les branches #2 et #3 ont actives, l'une recevant respectivement un courant de 600 nA et l'autre un courant de 300 nA. On fait varier le courant d'une troisième branche, la branche #1, pour vérifier le fonctionnement du 2-WsTA. Lorsque le courant d'entrée dans cette branche dépasse 300 nA, le courant de sortie n'est plus nul et cet élément s'active à la place d'un autre. De plus, on vérifie que

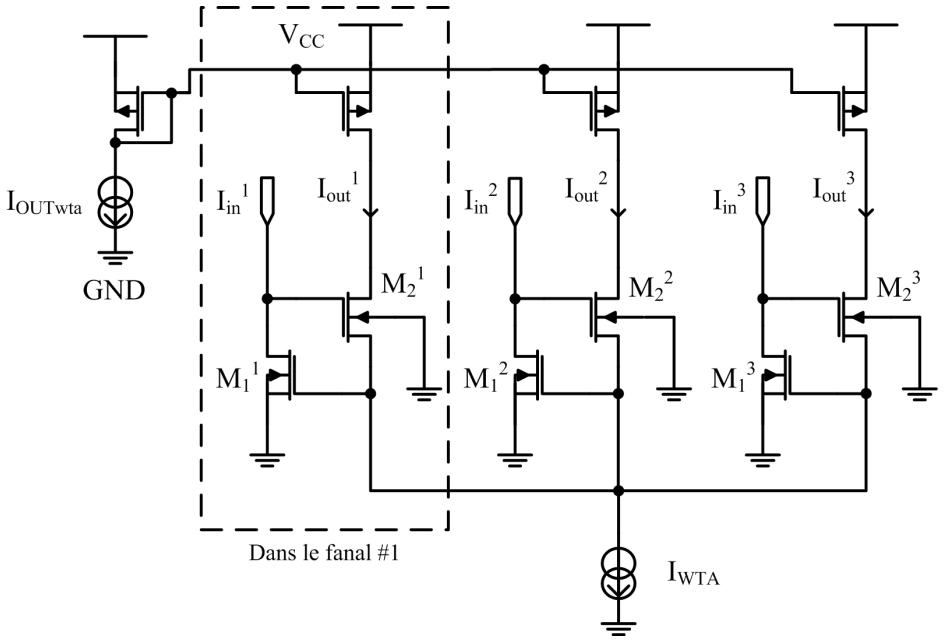

FIGURE 3.15 – Schéma électrique du circuit k-WsTA de [RH14].

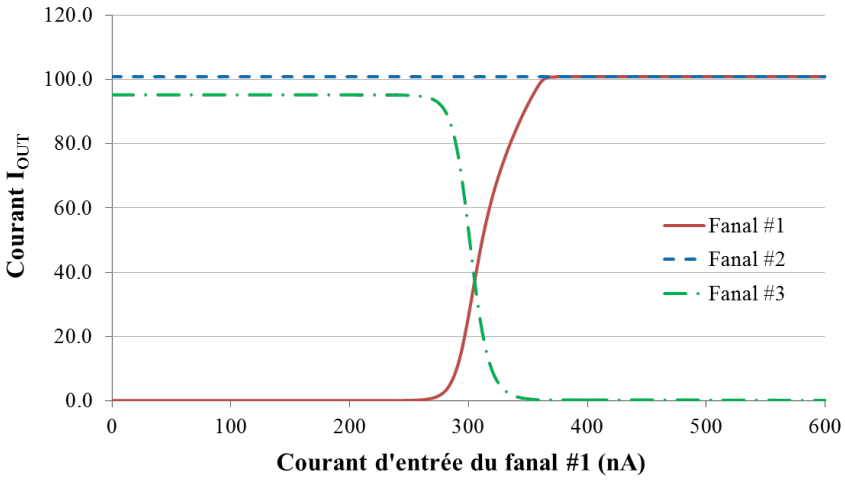

FIGURE 3.16 – Réponse du circuit 2-WsTA lors de la variation d'un courant entrant de 0 à 600nA.

la totalité des courants à tous les instants donne bien les 200 nA de la source I_{WTA}.

Le défaut de ce circuit concerne le cas où le nombre d'éléments à activer est supérieur à k. Ce cas peut arriver dans le cas de plusieurs fanaux ex-æquo pour la k-ième place. Si cela se produit, le courant se divise entre le fanaux ex-æquo et il passe un courant inférieur à $I_{OUT_{WTA}}$ dans les transistors M_2^i correspondants, ce qui gêne l'activation de ces éléments.

3.3.4 Opération "Losers-Kicked-Out"

L'opération "Losers-Kicked-Out" permet d'éliminer les fanaux stimulés qui ne sont pas dans la même clique que les autres fanaux stimulés. Cette opération est lourde et nécessite un processus en trois étapes. Contrairement aux autres règles de décodage, cette opération n'est pas faisable en continu, c'est-à-dire qu'un circuit implantant cette règle de décodage nécessite une partie commande cadençant le déroulement de chacune des étapes. En conséquence, une implantation totalement en électronique numérique se révèle beaucoup plus complexe qu'une autre règle de décodage, pour un gain de performance limité (voir Chapitre 1).

3.4 Décision sur le fanal actif

3.4.1 Prise de décision

Une fois que le circuit de comparaison a donné son résultat, le circuit de prise de décision a deux principaux objectifs : rendre *actifs* les fanaux dont les entrées satisfont la règle de comparaison, et fournir cette information sous la forme d'une tension binaire. Cette information sera ensuite transmise dans le réseau jusqu'à sa sortie, ainsi que vers d'autres fanaux.

Dans tous les circuits de comparaison étudiés précédemment, excepté le circuit WTA issu de

[DST98], dont la décision peut être prise de façon numérique via un circuit de retour, il est possible de prendre la décision grâce à une simple comparaison de la tension drain-source d'un transistor à son paramètre $V_{DS_{sat}}$. Afin de numériser le résultat grâce à une tension en sortie, cette comparaison à un seuil est réalisée par un buffer dont le seuil de basculement doit être supérieur à $V_{DS_{sat}}$ et inférieur à la tension V_{COMP} définie dans la section précédente.

3.4.2 Influence du nombre d'éléments de comparaison

Le circuit de décision est critique dans la réalisation du fanal, puisque c'est lui qui transmet l'état du fanal aux autres parties du réseau. Une mauvaise prise de décision peut donc impacter l'ensemble du décodage. En conséquence, certaines dispositions sont prises afin de protéger le buffer des conditions pouvant influer sur sa décision.

Dans le circuit WTA inspiré de [GC12], la valeur de la tension V_{COMP} est modifiée par la valeur du courant passant dans le transistor M_1^i, cf. Figure 3.12. Or, comme expliqué dans la Section 3.3.2.2, plus il y a d'éléments WTA en parallèle, plus un grand courant est prélevé du vainqueur pour passer dans les diodes bloquées en parallèle. La valeur de la tension V_{COMP} est donc conditionnée par le nombre d'éléments WTA en parallèle. Le seuil de décision doit donc être adapté en conséquence. Ainsi, dans le but d'avoir un circuit de fanal générique pour toutes les topologies de réseau, la fonction de pouvoir modifier le seuil de décision doit être mise en place. Pour cela, on limite le courant dans le premier des deux inverseurs composant le buffer de décision grâce à un transistor M_{21}, Figure 3.17.

Ceci permet de modifier son point de basculement grâce à une tension de commande $V_{commande}$, contrôlable extérieurement. La Figure 3.18 montre la valeur du point de basculement d'un tel circuit, en fonction de la tension $V_{commande}$. Avec ce circuit, le seuil de décision d'un élément WTA peut être indifféremment placé entre 45 mV et 487 mV, qui sont les valeurs extrêmes que peut

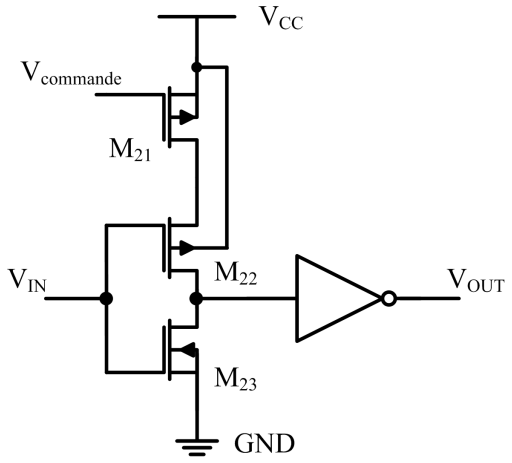

FIGURE 3.17 – Schéma électrique du buffer à seuil variable.

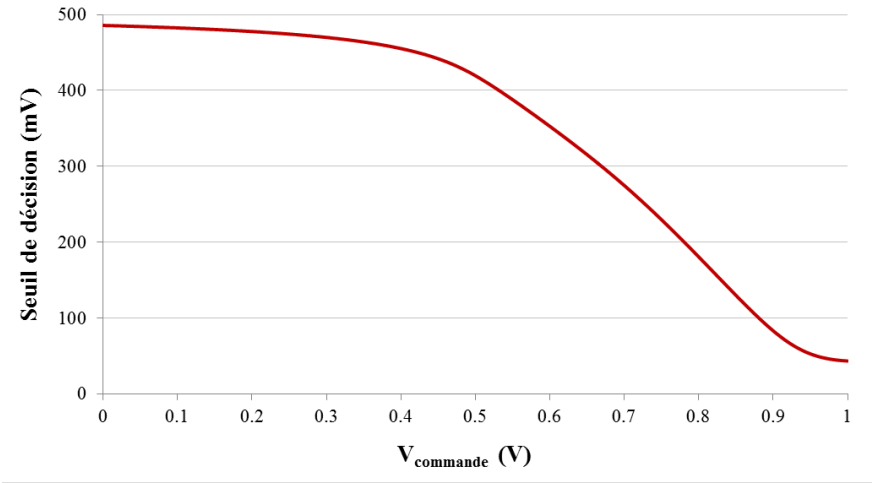

FIGURE 3.18 – Valeur du seuil de décision en fonction de $V_{commande}$, pour V_{CC}=1 V.

prendre le seuil de décision en faisant varier $V_{commande}$.

Cette solution permet donc de couvrir toute la plage de tension envisagée, mais elle nécessite un contrôle externe au réseau.

3.5 Réalisation d'un cluster de fanaux

3.5.1 Connexion des modules élémentaires

Les différents modules sont alors assemblés afin de former le fanal, représenté dans la Figure 3.19. Les synapses convertissent les informations fournies par les entrées en courants, qui sont ajoutés au point A. Le courant résultant est copié, puis envoyé dans la partie du circuit WTA présente dans un fanal. Une fois la comparaison réalisée, la décision se fait au point B par rapport à un seuil contrôlé par $V_{commande}$, image d'un seuil de référence V_{ref}. La sortie de ce buffer de décision donne l'état du fanal.

Le temps de réponse à une stimulation d'un fanal est donné par la Figure 3.20. Dans cette simulation, on stimule une entrée du fanal à t=5 ns. V_{CC} vaut 1 V, I_{UNIT} 300 nA et $V_{commande}$ 400 mV. Le fanal répond à cette stimulation en 7,3 ns.

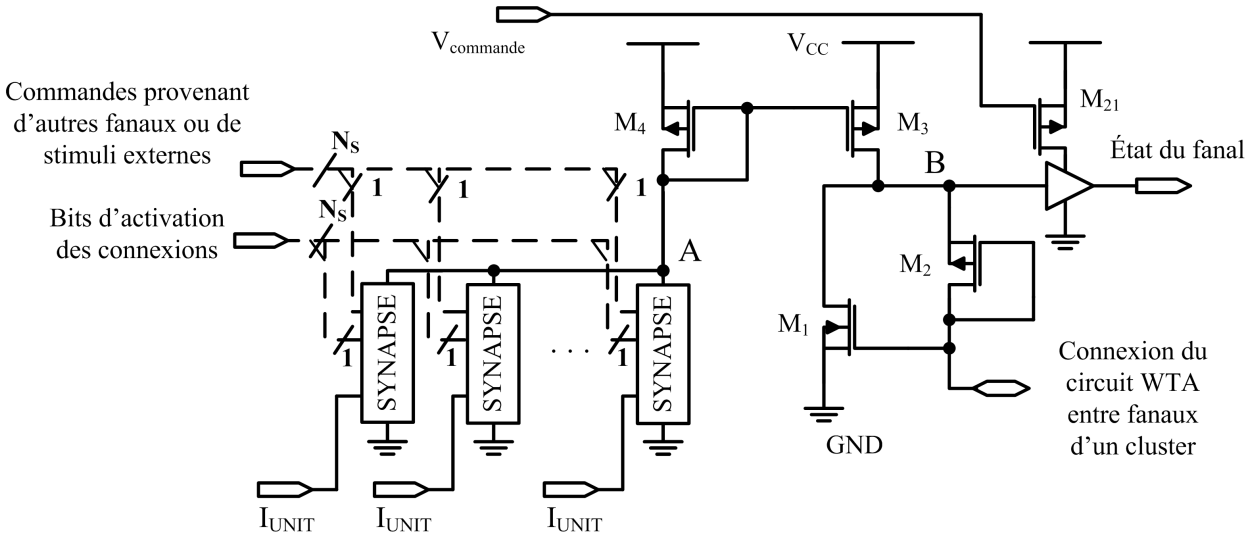

FIGURE 3.19 – Schéma électrique d'un fanal complet.

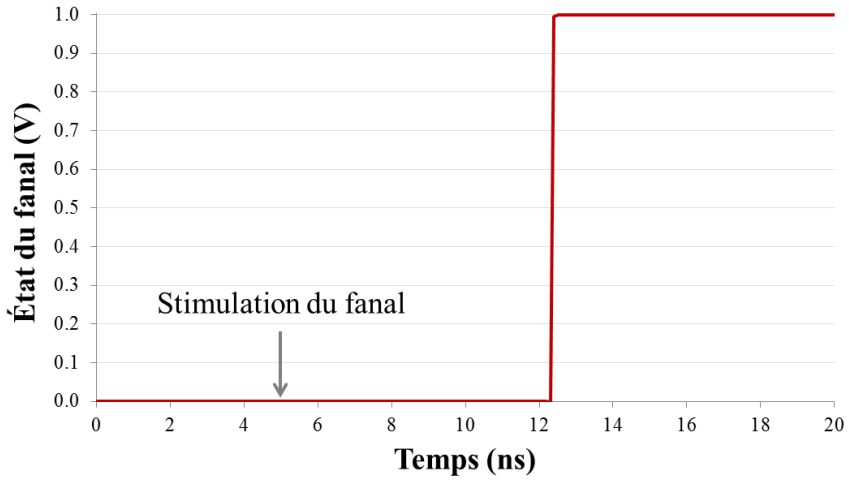

FIGURE 3.20 – Réponse du fanal à une unique stimulation, qui a lieu après 5 ns, dans différentes conditions environnementales.

3.5.2 Connexion des fanaux entre eux

Un cluster de fanaux est formé en reliant les éléments WTA entre eux, Figure 3.21. Cette connexion est réalisée au point C. De plus, la moitié du miroir servant à amener le courant I_{UNIT} dans une synapse, M_5, est commune à toutes les synapses du cluster, afin de réduire le nombre de transistors. La commande du seuil de décision $V_{commande}$ est également la même dans chaque cluster.

La Figure 3.22 montre le fonctionnement du cluster dans sa globalité. Dans un cluster de six fanaux initialement inactifs, le fanal #1 est stimulé après 5 ns. Ce fanal étant majoritaire, il s'active

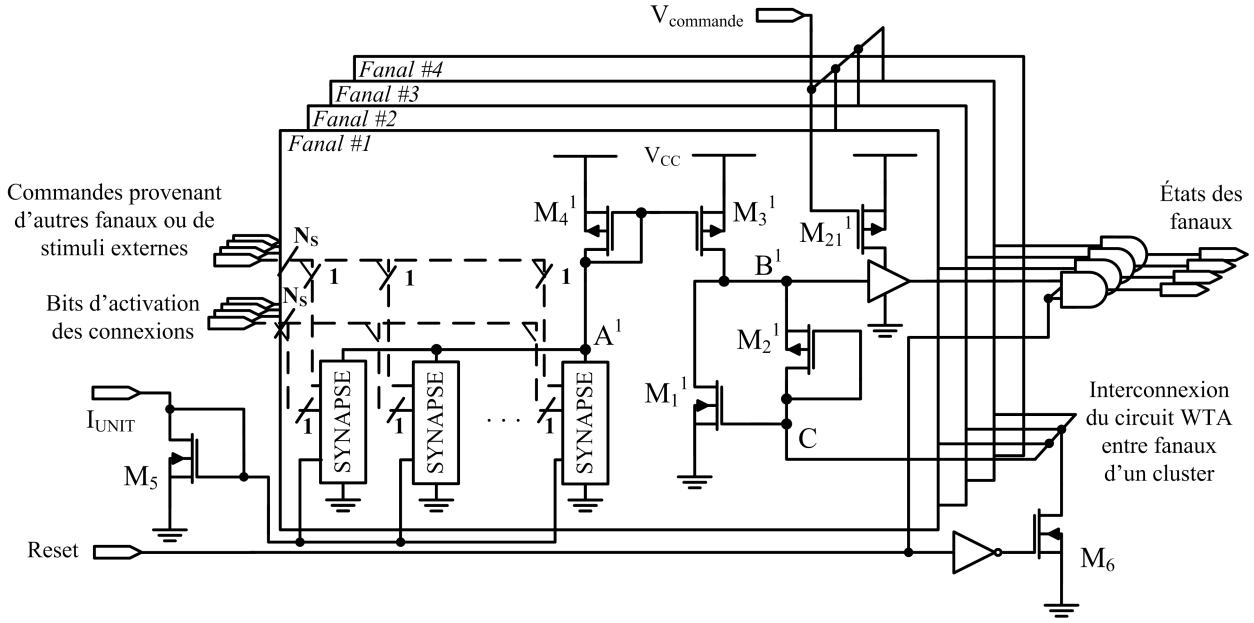

FIGURE 3.21 – Schéma électrique d'un cluster de quatre fanaux.

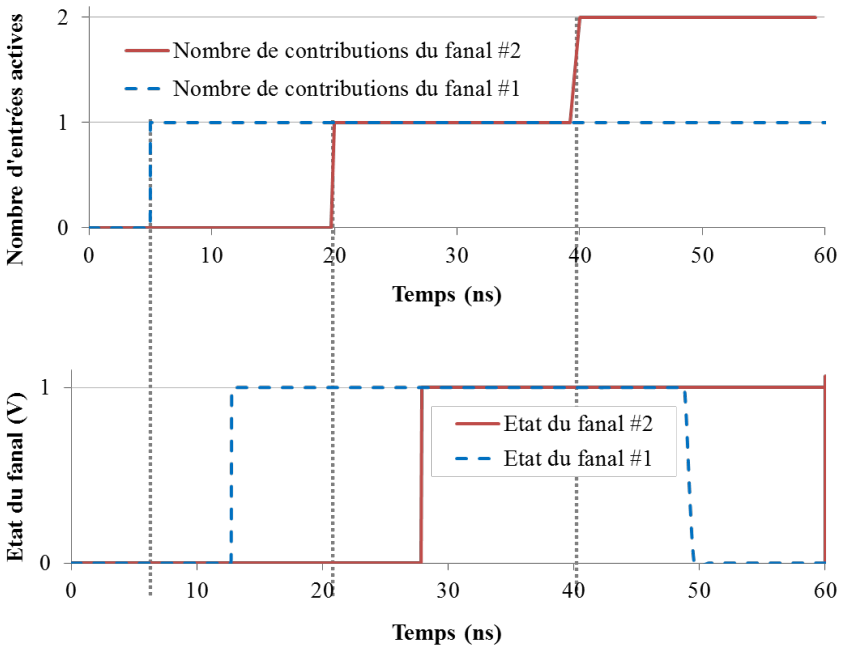

FIGURE 3.22 – Réponse des fanaux d'un cluster de six à différentes stimulations successives.

et tous les autres fanaux restent inactifs. Après 20 ns, le fanal #2 est stimulé à son tour. Les deux fanaux sont à égalité, le fanal #2 s'active donc et le fanal #1 reste actif. Enfin, après 30 ns de simulation, une autre entrée du fanal #2 est activée. Il devient majoritaire et le fanal #1 devient

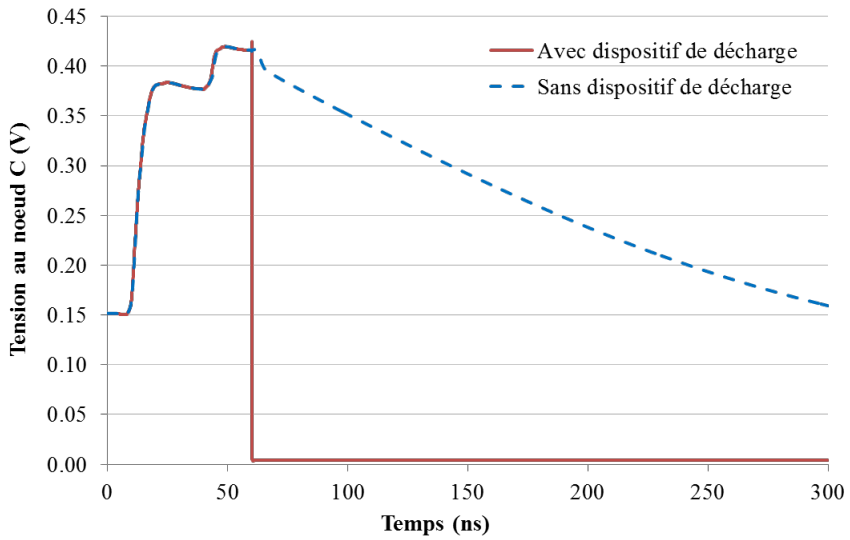

FIGURE 3.23 – Décharge du noeud C entre deux recherches de messages, avec et sans dispositif de décharge.

inactif. Le bon fonctionnement des fanaux est donc bien vérifié.

Finalement, un dispositif de remise à zéro de l'état des fanaux est mis en place, afin de réinitialiser toutes les sorties des fanaux du réseau pour démarrer la recherche d'un nouveau message. Une porte logique "ET" est donc placée après la sortie de chaque fanal. Elle effectue son opération entre la sortie du fanal et un signal global "Reset" actif à '0'. Ce signal est aussi utilisé pour décharger le nœud C après chaque recherche de message, grâce à un inverseur et au transistor M_6. En effet, ce nœud n'est connecté qu'à des grilles de transistors, et ne peut que se décharger lentement dans le substrat autrement. La dynamique de la remise à zéro des fanaux est montrée en simulation dans la Figure 3.23, avec et sans le dispositif de décharge du nœud C. L'ajout de ce dernier permet de diminuer le temps entre la fin d'une recherche de message et le début de la suivante de 400 ns.

Conclusion

Ce chapitre a montré qu'il est possible de réaliser les fonctions simples que nécessite le modèle de réseaux de neurones à cliques en électronique analogique. Le choix de faire les calculs en mode courant permet de s'affranchir de la complexité d'un additionneur et diminue la complexité du circuit de comparaison. En revanche, ce mode de calcul nécessite un étage de conversion dont la complexité est proportionnelle au nombre de connexions dans le réseau, ce qui augmente la taille du nœud de calcul.

Le chapitre suivant montre comment le nœud de calcul complet développé dans ce chapitre se comporte dans un réseau complet, et les performances attendues par rapport à la théorie.

Chapitre 4

Intégration analogique d'un réseau à cliques

Introduction

Dans le chapitre précédent nous avons présenté une implantation analogique du fanal. A partir d'une architecture définie dans le Chapitre 2, une implantation d'un réseau à cliques peut alors être proposée. L'objectif de ce chapitre est d'abord de dimensionner ce réseau, puis de simuler son fonctionnement. Nous définirons ainsi des métriques qui permettront d'évaluer les performances du circuit en fonction de sa taille. Ceci permettra de dimensionner au mieux le réseau à cliques. Un modèle comportemental sera également donné afin de pouvoir simuler le réseau entier dans un temps acceptable. Par la suite nous nous intéresserons aux effets liés au désappariement des transistors et effectuerons des simulations permettant d'évaluer la robustesse des circuits par rapport aux variations de technologie, de tension d'alimentation et de température, ou PVT pour "Process, Voltage and Temperature". Ceci nous conduira à apporter des modifications sur le circuit afin de prendre en compte ces effets.

4.1 Performances et dimensionnement du réseau à cliques

Afin de simuler électriquement le fonctionnement d'un réseau de neurones à cliques, il est nécessaire de fixer sa taille et le nombre de ses interconnexions. En effet, le dimensionnement physique du réseau influe sur plusieurs paramètres, comme les temps de réaction de chaque fonction dans les fanaux ainsi que les niveaux de tensions dans les éléments WTA par exemple, comme nous l'avons vu dans le Chapitre 3, Sections 3.2.3 et 3.3.2.2. Dans cette section nous commencerons par étudier l'impact de la taille du réseau et du nombre de ses interconnexions sur les performances (pouvoir de récupération d'information, temps de convergence, et consommation d'énergie) du circuit. Ceci nous conduira à proposer un réseau de cinq clusters comprenant six fanaux. Ce réseau sera étudié et intégré par la suite. Pour ce faire nous avons besoin d'un modèle comportemental permettant de simuler le circuit en un temps acceptable. Ce modèle sera décrit également dans cette section.

4.1.1 Métriques utilisées pour la caractérisation des performances du réseau

Les performances des circuits de réseaux à cliques peuvent être évaluées selon les trois critères suivants : le pouvoir de récupération d'information, le temps de convergence et la consommation d'énergie du réseau. Le premier critère représente la fonctionnalité du réseau. Les deux autres sont choisis car ils représentent les métriques permettant de caractériser un circuit, comme expliqué dans le Chapitre 2, Section 2.1.1. Nous ne nous intéressons pas à la surface de silicium dans cette section, car elle concerne la réalisation physique du circuit, décrite plus en détail dans le Chapitre 5. De plus, le débit de traitement est lié au temps de convergence et donc seul ce dernier est considéré.

4.1.1.1 Pouvoir de récupération d'information

Dans [GB11], le pouvoir de récupération d'information est évalué en représentant un taux d'erreurs de récupération en fonction du nombre de messages stockés dans le réseau. Cela donne aussi la quantité maximale d'information qui peut être stockée dans le réseau sans dégrader la récupération d'information. Cependant, une telle métrique n'est pas envisageable dans le cas de réseaux dont la taille et les interconnexions sont figées (c'est-à-dire réseau à dictionnaire fixe), car on ne peut pas faire varier le nombre de messages dans le dictionnaire. Dans ce cas là, il faut donc déterminer une autre représentation du pouvoir de récupération d'information. Cela peut être fait en observant le comportement du réseau lorsque l'intégralité des stimulations possibles sont présentées à l'entrée. Le pouvoir de récupération d'information est alors représenté par le taux de récupérations réussies en fonction du nombre d'erreurs insérées en entrée. Cette méthode, empruntée au domaine du codage correcteur d'erreur, donne la *trace* du réseau considéré.

La qualité de la récupération des messages est altérée quand la densité du réseau augmente, comme décrit dans [GB11]. En effet, si la densité de connexions – et donc le nombre de connexions – augmente, l'ensemble des connexions peut former des cliques non désirées. La probabilité que

ces cliques, appelées *cliques parasites*, se forment augmente avec la densité de connexions. Ces cliques altèrent la récupération des messages dans certains cas de stimulation. De plus, une densité importante diminue aussi la distance de Hamming entre les mots du dictionnaire, c'est-à-dire le nombre de lettres minimal différentes d'un mot du dictionnaire à un autre. La densité maximale que permet le réseau à cliques sans altérer la récupération d'information est d'environ 25%, d'après [GB11]. Cette limite est donnée dans le cas d'un stockage uniforme des messages, c'est-à-dire où tous les fanaux du réseau ont environ le même degré de connectivité.

4.1.1.2 Temps de convergence

Le temps de convergence du réseau T_{CONV} est défini comme le plus long temps nécessaire au réseau de neurones pour converger sur une décision de récupération parmi toutes les récupérations possibles. Comme le fonctionnement du réseau se base sur l'itération du processus de récupération de messages décrit dans le Chapitre 1, Section 1.3.3.1, l'arrêt de la récupération d'un message doit être imposé depuis l'extérieur du réseau au bout d'un certain nombre d'itérations de ce processus, correspondant à une durée T_{CONV}. Ce temps de convergence est donc un paramètre déterminant car si la décision est prise trop tôt, la convergence peut ne pas être atteinte dans tous les cas.

Le temps de convergence dépend de deux paramètres du circuit : le nombre de synapses et le nombre d'éléments WTA en parallèle. Comme montré dans la Section 3.2.3, l'augmentation du nombre de synapses dans un fanal augmente le temps de réaction de l'additionneur de 100 ps par synapse. Or, d'après l'équation (2.2) de la Section 2.2.1, tous les paramètres topologiques (nombre de clusters, nombre de fanaux par cluster et densité du réseau) influent sur le nombre de synapses.

De plus, d'après la Section 3.3.2.2, l'augmentation du nombre d'éléments WTA en parallèle augmente le temps de réaction du circuit WTA de 120 ps par élément WTA rajouté en parallèle. Le nombre d'éléments WTA en parallèle est lui directement lié au nombre de fanaux par cluster.

D'une manière générale, l'augmentation d'un des trois paramètres topologiques étudiés augmente le temps de réponse du réseau.

4.1.1.3 Consommation d'énergie

La consommation d'énergie du réseau est également considérée. Elle se divise en deux parties. Tout d'abord, nous définissons la consommation statique comme la consommation en dehors des périodes de récupération. Elle est donc essentiellement liée à la polarisation et est partagée par tous les fanaux dans le réseau à cliques. Ensuite, durant les phases de récupération, le réseau consomme une quantité d'énergie supplémentaire que nous définissons comme la consommation dynamique. Cette dernière peut donc être imputée aux fanaux actifs durant la récupération. La consommation totale du réseau sera donc différenciée entre les fanaux actifs (consommation statique et dynamique) et inactifs (consommation statique).

La consommation du réseau est liée aux nombre de fanaux du réseau. La consommation statique liée aux courants de fuite dans ces éléments est donc proportionnelle au nombre de fanaux dans le

réseau en entier. De plus, la consommation du réseau pendant la récupération d'un message dépend du nombre de fanaux actifs, donc du nombre de clusters, et de la densité du réseau, car plus de courant circule alors dans les synapses actives.

4.1.2 Dimensionnement du réseau

Les circuits implantant les fonctions élémentaires (additionneur, WTA, module de décision) présentés dans le Chapitre 3 ne sont pas idéaux : ils contiennent des éléments passifs parasites et sont sujets à des courants de fuite. L'intégration en parallèle d'un grand nombre de fanaux et de synapses amplifie les effets de ces éléments indésirables et peut conduire à un mauvais fonctionnement du circuit.

La topologie utilisée dans le réseau intégré sur puce doit respecter les limites exposées précédemment. La puce étant une preuve de concept de l'implantation du modèle, présentée dans le Chapitre 3, un réseau simple est choisi pour être intégré à l'intérieur, afin de valider le fonctionnement du circuit. Le réseau doit contenir suffisamment peu de fanaux pour que le dessin des masques soit aisé à concevoir de manière "full-custom". Mais il doit aussi en contenir assez pour que ses performances puissent être extrapolées à un réseau plus grand. Un bon compromis est un réseau de l'ordre de la dizaine de fanaux. Ainsi, un réseau de cinq clusters de six fanaux chacun respecte les

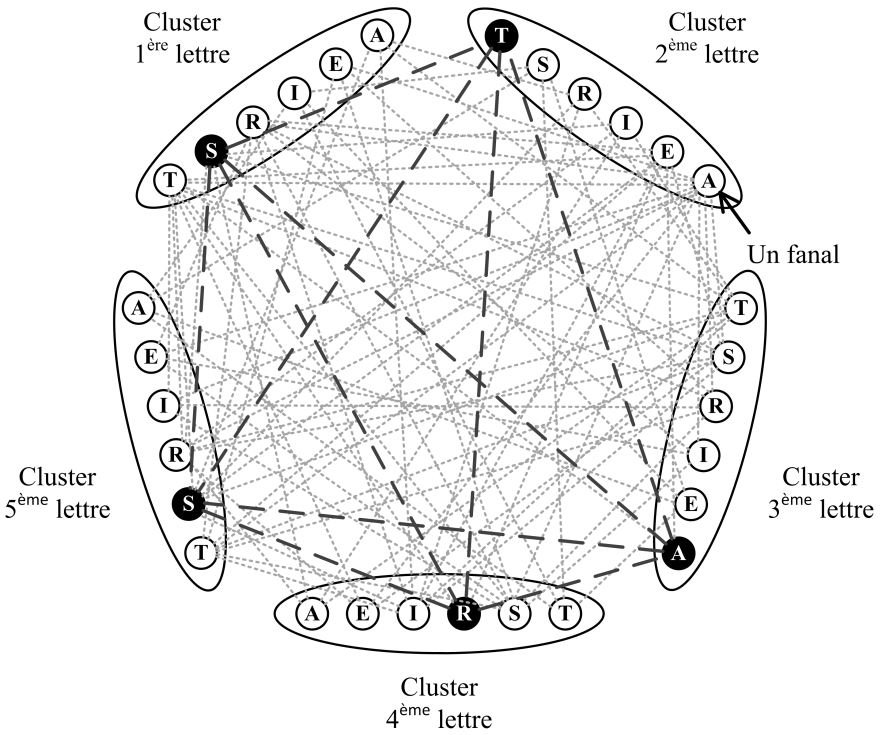

FIGURE 4.1 – Schéma du réseau à cliques considéré, constitué de cinq clusters de six fanaux chacun. Dix mots sont stockés dans ce réseau, et le mot "STARS" est mis en évidence.

limitations et est assez simple pour valider le fonctionnement du circuit. Ce réseau représente le stockage de mots de cinq lettres composés à partir d'un alphabet restreint à six lettres, comme montré sur la Figure 4.1. L'alphabet considéré est composé des lettres A, E, I, R, S et T.

De plus, la méthode de communication proposée dans la Section 2.2.3.1 est envisagée. Il s'agit de connecter les fanaux entre eux grâce à de simples liens directs métalliques. Le nombre de fanaux est suffisamment bas pour que cette méthode de communication soit plus avantageuse que celles définies dans la Section 2.2.3.2. En revanche, une fois le dictionnaire établi, il ne sera plus possible de le modifier.

Afin de stocker le maximum d'informations possible, dix mots forment le dictionnaire stocké dans ce réseau. Ceux-ci sont choisis de façon à ce que tous les fanaux appartiennent à au moins un mot. Des mots ayant jusqu'à trois lettres en commun sont aussi choisis, pour montrer l'effet d'un apprentissage non-uniforme sur les performances de récupération [BGSH14]. Le dictionnaire complet est donné dans le Tableau 4.1. La densité du réseau est alors de 26%, soit à la limite de la densité maximale acceptée. De cette manière, le pouvoir de récupération n'est pas altéré par la présence de cliques parasites.

TABLEAU 4.1 – Dictionnaire des mots stockés dans le réseau.

STARS	SITAR
ARTIS	TARSI
RAISE	RESIT
TEARS	EARST
TASER	ISETA

4.1.3 Modélisation comportementale du circuit d'un réseau complet

Afin d'évaluer les performances du réseau à cliques considéré, il est nécessaire de réaliser les simulations correspondant à toutes les stimulations possibles pour le réseau, soit 16 807 simulations. Or, une simulation au niveau transistor du réseau considéré, cinq clusters de six fanaux, prend quelques dizaines de secondes. La réalisation de toutes ces simulations prend donc plus de six jours si elles sont effectuées au niveau du transistor. De plus, pour des réseaux plus grands, le nombre de transistors dans le réseau va augmenter de manière quadratique en fonction du nombre de fanaux total, comme montré dans la Section 2.2.3.2. La durée de chaque simulation va donc s'allonger fortement, comme le temps de simulation total.

Il faut donc réaliser ces simulations à un plus haut niveau, où moins de calculs pour le simulateur sont nécessaires. Un modèle comportemental du circuit du réseau à cliques est alors réalisé avec *Simulink*®. Chaque fonction élémentaire, c'est-à-dire addition des contributions, comparaison et

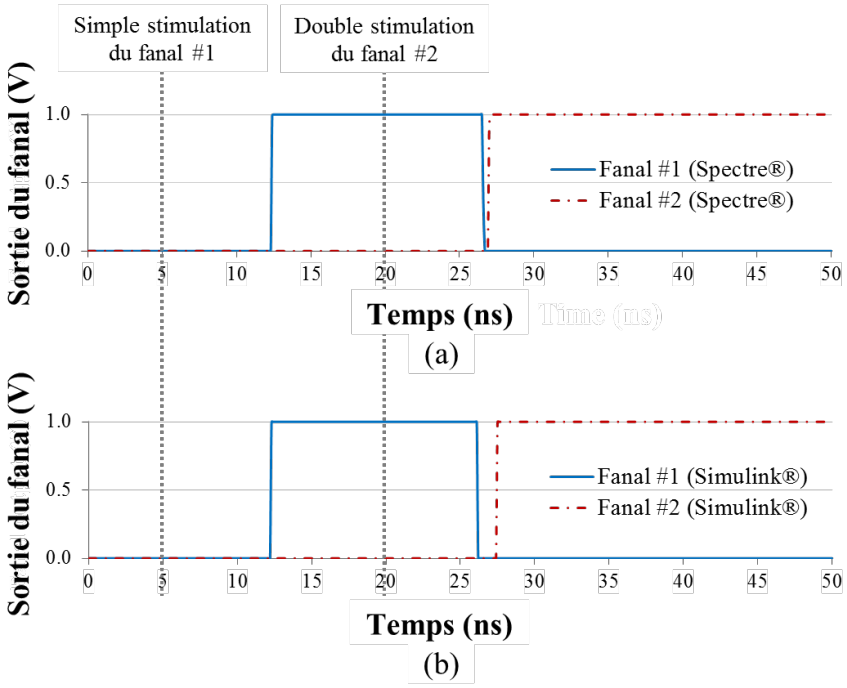

FIGURE 4.2 – Résultats de simulations concernant le fonctionnement du réseau dans le cas de la correction d'une erreur. Les réponses de deux fanaux sont montrées pour des simulations utilisant *Spectre*® (a) et *Simulink*® (b). Une synapse d'un fanal, le fanal #1, est tout d'abord stimulée après 5 ns de simulation, puis deux synapses d'un autre fanal appartenant au même cluster, le fanal #2, sont stimulées après 20 ns de simulation.

décision, est réalisée grâce à un bloc écrit en langage C. Ces blocs sont alors connectés grâce à des lignes de transmission modélisant les lignes électriques. Celles-ci sont modélisées en se basant sur les temps de propagation obtenues lors des simulations au niveau du transistor du circuit.

Avant d'utiliser ce modèle pour caractériser le réseau considéré précédemment, il convient de vérifier sa validité sur un cas de simulation simple. Pour cela, on considère le cas d'une correction d'erreur. Les réponses des fanaux simulées avec *Spectre*® et avec *Simulink*® sont montrées sur la Figure 4.2. Une synapse d'un fanal, le fanal #1, est tout d'abord stimulée après 5 ns de simulation, puis deux synapses d'un autre fanal appartenant au même cluster, le fanal #2, sont stimulées après 20 ns de simulation.

Le fanal #1 est activé après 7,3 ns après la première stimulation, puis désactivé au profit du fanal #2. Le résultat final est obtenu après 7,5 ns après la seconde stimulation. Le fonctionnement des fanaux simulés avec *Simulink*® correspond aux simulations effectuées avec *Spectre*®. En revanche, les temps de réponse des fanaux ne sont pas identiques, mais cette différence est faible par rapport aux valeurs des temps de réponse des fanaux, et n'influe donc pas sur leur comportement. Ce modèle comportemental pourra donc être utilisé pour caractériser le réseau considéré.

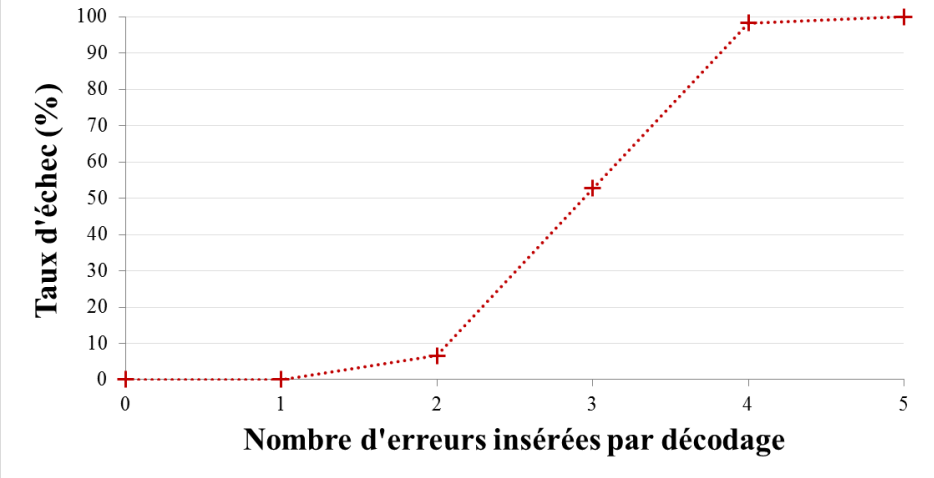

FIGURE 4.3 – Taux d'erreur du réseau en fonction du nombre d'erreurs introduites dans la stimulation. Les tests sont réalisés en simulation avec *Simulink*®.

4.1.4 Réponse du réseau en simulation

Les performances du réseau à cliques composé de cinq clusters de six fanaux sont alors évaluées en utilisant le modèle comportemental du circuit. La trace du réseau est montrée sur la Figure 4.3.

Sans erreur introduite dans la stimulation, le réseau retrouve le bon message dans tous les cas. Avec une erreur introduite, ce pourcentage est réduit à 97%, 86% pour deux erreurs, 42% pour trois erreurs, 3% pour quatre erreurs, et 0,03% pour un mot stimulé complètement erroné. On pourrait s'attendre à un meilleur pourcentage de récupérations réussies dans les cas contenant une ou deux erreurs de stimulation, proche de 0%, les contributions des fanaux correctement activés corrigeant théoriquement les erreurs. Or, la densité du réseau proche de la limite théorique combinée au stockage de données non uniforme dégrade nettement certaines récupérations.

De plus, le temps de convergence du réseau est de 21 ns, et sa consommation simulée avec *Spectre*® est de 3,8 μW par fanal inactif et de 6,9 μW par fanal actif, soit 129,5 μW au total.

Cependant, ces performances sont données pour des conditions de fonctionnement idéales. Or, dans un circuit intégré, les conditions de fonctionnement, comme la température, la tension d'alimentation peuvent varier localement. De même, des défauts de fabrication, au niveau du dopage du substrat ou des tailles des transistors peuvent être présents. Toutes ces imperfections peuvent altérer le fonctionnement du circuit localement, et peuvent donc avoir un effet sur le réseau dans son ensemble. Les sections suivantes vont donc vérifier le comportement du réseau en présence de ces imperfections et, si besoin, proposer des solutions pour minimiser leur impact sur le fonctionnement du circuit.

4.2 Appariement des transistors

4.2.1 Effet des problèmes d'appariement sur le circuit

Un problème pouvant se poser dans les circuits analogiques, et particulièrement dans les circuits basés sur la recopie de courants, est issu des défauts d'appariement des transistors. Le défaut d'appariement des transistors est dû à de faibles différences de taille entre les transistors appairés, ou à une variation de la tension de seuil. Ces problèmes affectent les miroirs de courants et les comparateurs, et sont particulièrement important pour des transistors de taille minimale. En conséquence, les courants de sortie des miroirs de courants n'ont pas exactement la valeur attendue, et les comparateurs peuvent avoir des effets d'offset non désirés, [PDW89]. Cela fausse certaines comparaisons, et diminue ainsi le taux de récupérations de messages réussies.

Il est donc important de prévoir quels peuvent être les effets de ces défauts sur les éléments composant le fanal, ainsi que leur impact sur le fanal en entier, puis sur le réseau à cliques complet.

4.2.1.1 Effet des problèmes d'appariement sur les synapses

La génération du courant unitaire dans la synapse étant faite par un seul miroir de courant, celle-ci est affectée par le désappariement des transistors. Ces défauts ont pour conséquence un étalement de la valeur du courant de sortie, et donc une modification du caractère binaire de la synapse.

Des simulations de type Monte-Carlo effectuées à l'aide de *Spectre*® permettent d'estimer cet

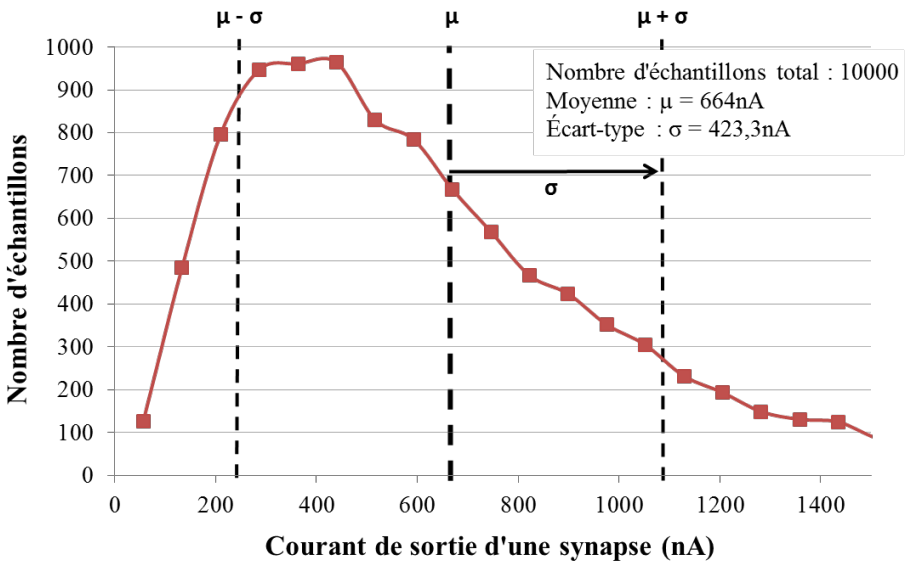

FIGURE 4.4 – Simulation Monte-Carlo donnant l'étalement de la valeur du courant de sortie d'une synapse composée de transistors de longueurs L_{min}, sur 10 000 échantillons, et pour V_{CC}=1 V et I_{UNIT}=300 nA.

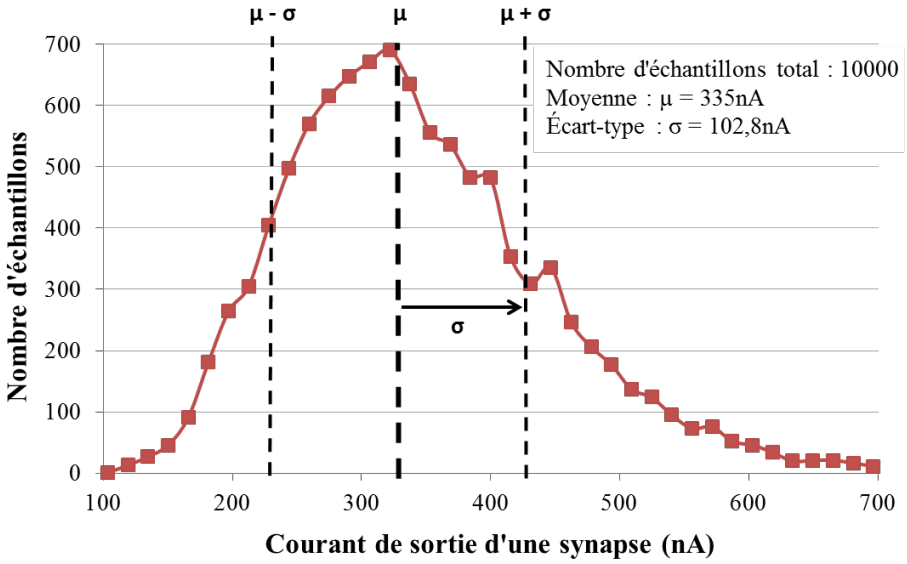

FIGURE 4.5 – Simulation Monte-Carlo donnant l'étalement de la valeur du courant de sortie d'une synapse composée de transistors de longueurs $3L_{min}$, sur 10 000 échantillons, et pour V_{CC}=1 V et I_{UNIT}=300 nA.

étalement de valeurs. Les paramètres des transistors MOS varient alors aléatoirement, simulant un défaut d'appariement. Ces simulations sont répétées un grand nombre de fois, afin d'avoir une distribution statistique significative des résultats obtenus. La Figure 4.4 montre une telle simulation effectuée au niveau de la sortie d'une synapse. Elle indique ainsi la moyenne μ et la valeur de l'écart-type σ de la distribution du courant que l'on peut espérer lors du fonctionnement. On peut remarquer que la distribution de courant est dissymétrique, et donc que μ est éloigné de I_{UNIT}. De plus, on remarque aussi que σ est bien supérieur à I_{UNIT}, ce qui implique qu'une synapse peut très bien délivrer l'équivalent d'aucune ou de plusieurs contributions. Ceci peut avantager ou non un fanal par rapport à un autre pour le même nombre de contributions.

Cependant, cet écart-type est inversement proportionnel à $\sqrt{WL}$ [CRD+02], où W et L sont respectivement la largeur et la longueur de canal du transistor MOS. On peut donc augmenter les dimensions des transistors afin de limiter l'impact du défaut d'appariement des transistors, comme montré sur la Figure 4.5. La moyenne de la distribution de courant se rapproche de la valeur de I_{UNIT}, et l'écart-type est bien plus faible. Des techniques de dessin des masques, comme l'utilisation de grilles factices autour du transistor, permettent aussi de diminuer cet étalement de courant.

Si l'impact de cet étalement du courant est visible en sortie de la synapse, il n'est pas évident d'estimer quel est son impact sur un réseau complet. Selon le nombre de synapses actives et la topologie du réseau, la redondance de l'information dans le réseau peut être suffisante pour compenser les éventuelles disparités entre fanaux. Des simulations à l'échelle du réseau sont donc nécessaires pour estimer l'effet du désappariement des transisors dans les synapses.

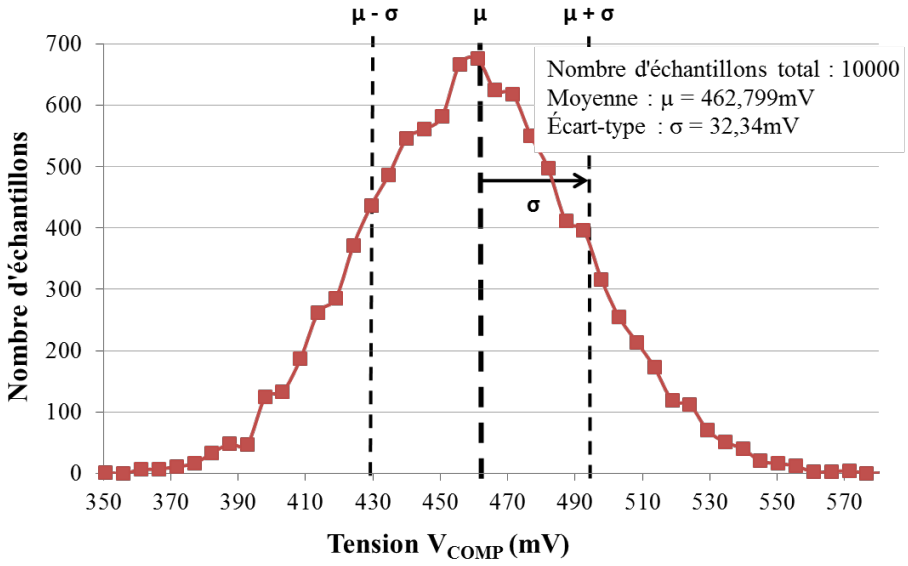

FIGURE 4.6 – Simulation Monte-Carlo donnant l'étalement de la tension V_{COMP}, sur 10 000 échantillons, et pour V_{CC}=1 V et I_{UNIT}=300 nA.

4.2.1.2 Effet des problèmes d'appariement sur le circuit WTA

Le défaut d'appariement des transistors a aussi un effet sur l'élément WTA. Le point qui peut poser problème au niveau de cet élément est la modification de la valeur de la tension V_{COMP}, susceptible d'être affectée par un défaut de M_1 (sur le schéma de la Figure 3.19 du Chapitre 3, Section 3.5.1).

Des simulations Monte-Carlo ont aussi été réalisées sur le circuit WTA afin d'estimer la variation due au défaut d'appariement des transistors. La Figure 4.6 donne la dispersion possible de la tension V_{COMP}. On s'aperçoit qu'elle est d'environ 30 mV, autour d'une tension moyenne de 462 mV.

Ces variations ne remettent pas en cause la fonctionnalité du circuit WTA, puisque la différence avec un élément inactif est toujours significative, et simulée à environ 300 mV.

4.2.1.3 Effet des problèmes d'appariement sur le circuit de décision

Le seuil de décision peut aussi être modifié par les défauts d'appariement des transistors. La Figure 4.7 montre une simulation Monte-Carlo sur la valeur du seuil de décision, fixée en théorie à 400 mV, en prenant en compte les défauts d'appariement uniquement. L'écart-type est d'environ 10 mV, et vient s'ajouter à l'écart dû aux variations des paramètres environnementaux. Il est donc possible de corriger ce défaut par le dispositif de compensation présenté dans la Section 4.3.3.

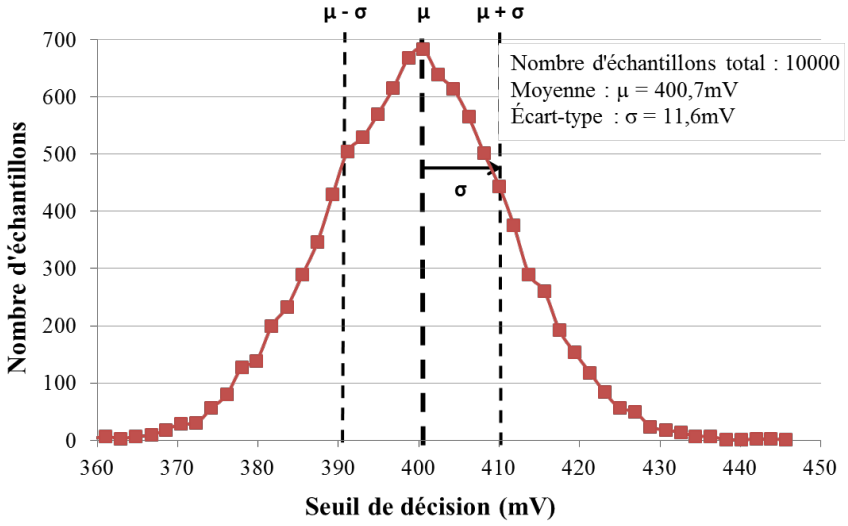

FIGURE 4.7 – Simulation Monte-Carlo donnant l'étalement de la valeur du seuil de décision, sur 10 000 échantillons et pour V_{CC}=1 V

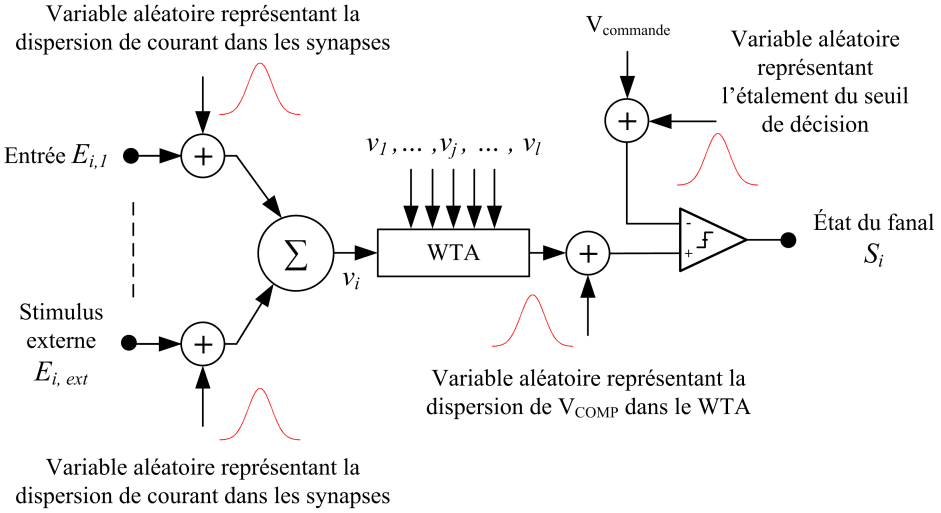

FIGURE 4.8 – Schéma du fanal modélisé sous *Simulink*®, incluant les effets de désappariement des transistors.

4.2.2 Impact sur la récupération des messages

Comme pour l'effet des paramètres environnementaux, on peut simuler l'impact du défaut d'appariement des transistors sur le réseau à cliques complet grâce au modèle comportemental du réseau. Avec le modèle comportemental idéal décrit en Section 4.1.4, les contributions fournies par chaque synapse à l'additionneur sont fixes. Il n'y a donc pas de problèmes de recopie de courant dans les miroirs.

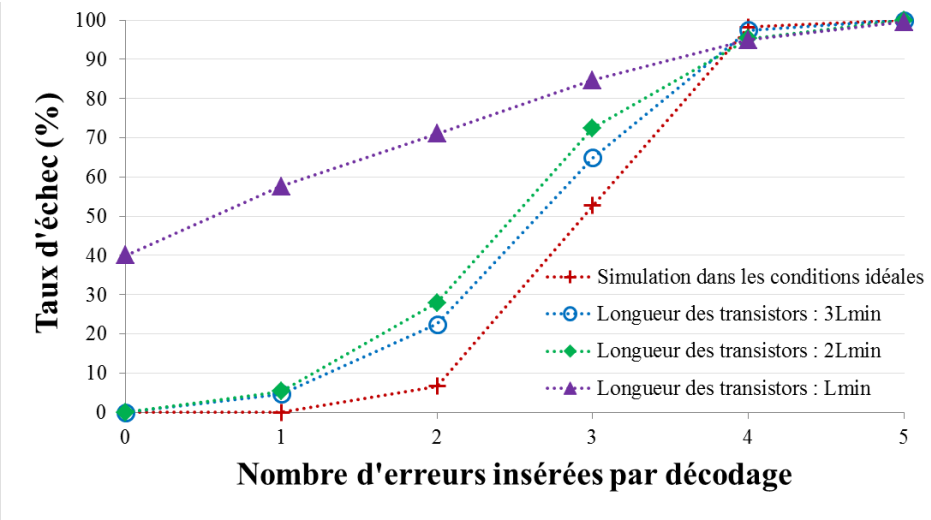

FIGURE 4.9 – Taux d'erreur du réseau en fonction du nombre d'erreurs introduites dans la stimulation. Les tests sont réalisés en simulation *Simulink*®, avec et sans défaut d'appariement des transistors.

Ces valeurs fixes (seuils et courants) sont remplacées par des variables aléatoires indépendantes les unes des autres dans le modèle, comme le montre la Figure 4.8. Les distributions de ces variables aléatoires sont conformes aux simulations Monte-Carlo des Figures 4.4, 4.5, 4.6 et 4.7. Ce modèle permet ainsi de simuler des défauts d'appariement correspondant à différentes tailles de transistors.

La trace du réseau à cliques sujet à des défauts d'appariement des transistors est donnée sur la Figure 4.9, pour différentes valeurs de longueur des transistors. Les simulations sont effectuées pour V_{CC}=1 V, I_{UNIT}=300 nA et $V_{commande}$=400 mV. Ces traces sont des moyennes de vingt calculs de traces, avec un nouveau tirage des variables aléatoires à chaque calcul. Les longueurs de transistors choisies pour la simulation sont la longueur minimale disponible dans le kit L_{min}, $2L_{min}$ et $3L_{min}$.

Nous pouvons remarquer une nette amélioration des performances de récupération quand on passe d'une longueur de L_{min} à $2L_{min}$ (réduction maximale du taux d'erreur de 52% pour une erreur stimulée). Ce gain est moins net lorsque l'on passe de $2L_{min}$ à $3L_{min}$, avec au maximum 21% pour deux erreurs stimulées. D'une manière générale, un gain de performances au niveau du pouvoir de récupération est obtenu au prix d'une plus grande surface occupée par les transistors, mais ces derniers sont plus robustes au défaut d'appariement. Un compromis doit alors être trouvé lors de la conception du circuit, en privilégiant soit la fonctionnalité, soit la compacité du circuit.

Dans le cas de cette étude, on choisit de maximiser la robustesse des transistors afin de concevoir une puce dont le but est de montrer les performances de récupération de messages du réseau à cliques. On utilisera donc des transistors de longueur $3L_{min}$ dans le circuit, afin de minimiser les effets des défauts d'appariement des transistors dans le réseau. Dans la section suivante, nous nous intéressons aux variations des conditions environnementales dans le circuit.

4.3 Compensation des variations des paramètres environnementaux

4.3.1 Effets des paramètres environnementaux sur le circuit

Une source d'imprécisions potentielle dans le fonctionnement d'un circuit est la variation des paramètres de l'environnement que sont les variations de technologie, de température et de tension d'alimentation. Les variations de ces paramètres sont dues à des gradients de température, dopage du substrat, ou encore de pertes en ligne sur un circuit intégré. Une modification dans l'un ou plusieurs de ces paramètres peut en effet changer le point de fonctionnement d'un circuit, et donc remettre en cause une ou plusieurs de ses fonctionnalités.

Des simulations incluant ces données peuvent alors être effectuées afin de vérifier la robustesse du circuit dans des conditions non-idéales. Nous choisissons une variation extrême de la tension d'alimentation de plus ou moins 10%, soit dans ce cas V_{CC} compris entre 0,9 V et 1,1 V. Une variation de 10% de la tension d'alimentation est déjà significative. Les températures extrêmes de fonctionnement sont fixées à 0 ° C et 80 ° C. Les conditions extrêmes de variations de technologie sont fournies par le fondeur, ce sont les variations *FF* et *SS* (*F* signifie *FAST*, *S* signifie *SLOW* et *T* signifie *TYPICAL*, la première lettre est pour les transistors NMOS et la seconde pour les transistors PMOS). Nous pouvons donc simuler le circuit dans les conditions *TYPICAL*, c'est-à-dire le cas typique, et les deux cas extrêmes. Les conditions *FAST* rassemblent les conditions qui maximisent la vitesse des porteurs de charge, tandis que les conditions *SLOW* rassemblent les conditions qui la minimisent.

4.3.1.1 Effets des paramètres environnementaux sur les synapses

Les variations des paramètres environnementaux définis précédemment peuvent avoir un impact sur les synapses. Dans ce cas, la valeur du courant en sortie d'une synapse peut différer suivant les conditions dans lesquelles celle-ci se trouve. Il faut donc évaluer l'écart entre la valeur du courant provenant de synapses dans les conditions extrêmes, c'est-à-dire *FAST* et *SLOW*, et la valeur typique de ce courant. Pour cela, des simulations sont effectuées avec *Spectre*®. Pour une tension d'alimentation V_{CC} de 1 V et une valeur de courant unitaire I_{UNIT} de 300 nA, la valeur du courant en sortie d'une synapse est simulée, pour chacune des conditions *TYPICAL*, *FAST* et *SLOW*. Le Tableau 4.2 recense les valeurs du courant de sortie dans ces trois cas.

Nous remarquons que les variations des paramètres environnementaux n'ont pas d'effet sur la valeur du courant de sortie d'une synapse. La grande longueur $3L_{min}$ des transistors les rend moins sensibles aux variations de technologie. De plus, la tension drain-source minimale pour que les transistors des synapses soient saturés, c'est-à-dire en fonctionnement normal, est d'environ 90 mV, donc très inférieure à la tension d'alimentation. Des variations de 100 mV de la tension d'alimentation n'ont donc pas d'impact sur le fonctionnement d'une synapse.

TABLEAU 4.2 – Valeurs du courant en sortie d'une synapse en fonction des conditions environnementales.

Valeur des paramètres environnementaux	Courant de sortie de la synapse
Conditions *"TYPICAL"* (technologie TT, T=27 ° C, V_{CC}=1 V)	303 nA
Conditions *"FAST"* (technologie FF, T=0 ° C, V_{CC}=1,1 V)	305 nA
Conditions *"SLOW"* (technologie SS, T=80 ° C, V_{CC}=0,9 V)	302 nA

4.3.1.2 Effets des paramètres environnementaux sur le circuit WTA

L'élément WTA peut aussi être un élément sensible aux variations des paramètres environnementaux. Ces derniers ont un effet sur la tension V_{COMP}, image du résultat de la comparaison WTA (cf. Figure 3.12 du Chapitre 3, Section 3.3.2.1). Les valeurs de V_{COMP} aux états "actif" et "inactif", c'est-à-dire si l'élément WTA est l'élément vainqueur ou non, sont simulées dans les mêmes conditions que dans la section 4.3.1.1. Le Tableau 4.3 donne la valeur de V_{COMP} dans les différents cas étudiés pour la variation des paramètres environnementaux.

TABLEAU 4.3 – Valeurs de la tension V_{COMP} en fonction des conditions environnementales.

Valeur des paramètres environnementaux	Tension V_{COMP} à l'état "actif"	Tension V_{COMP} à l'état "inactif"
Conditions *"TYPICAL"* (technologie TT, T=27 ° C, V_{CC}=1 V)	449 mV	41 mV
Conditions *"FAST"* (technologie FF, T=0 ° C, V_{CC}=1,1 V)	435 mV	39 mV
Conditions *"SLOW"* (technologie SS, T=80 ° C, V_{CC}=0,9 V)	454 mV	47 mV

Nous pouvons ainsi espérer au maximum une variation de 14 mV pour la tension V_{COMP} à l'état "actif" entre le cas typique et les cas extrêmes. Cette valeur semble faible, et nous allons vérifier si ces variations ont un impact sur le fonctionnement du fanal.

4.3.1.3 Effets des paramètres environnementaux sur la prise de décision

Les conditions environnementales (technologie, température et tension d'alimentation), ainsi que les imperfections dans les tailles des transistors ont pour effet de modifier la valeur du seuil de décision. Ceci entraîne des disparités en terme de réponse des fanaux, et donc des erreurs se propageant potentiellement plus rapidement que le message original. Afin d'étudier l'impact de ces paramètres sur le buffer de décision, des simulations faisant varier les paramètres environnementaux sont réalisées. Le seuil de décision visé est de 350 mV. Les deux cas de conditions extrêmes, les corners *"FAST"* et *"SLOW"*, et les conditions typiques, c'est-à-dire le corner *"TYPICAL"*, sont considérés. Le Tableau 4.4 donne les valeurs du seuil de décision pour chacune des conditions.

TABLEAU 4.4 – Valeurs du seuil de décision en fonction des conditions environnementales.

Valeur des paramètres environnementaux	Seuil de décision du buffer
Conditions *"TYPICAL"* (technologie TT, T=27 ° C, V_{CC}=1 V)	350 mV
Conditions *"FAST"* (technologie FF, T=0 ° C, V_{CC}=1,1 V)	407 mV
Conditions *"SLOW"* (technologie SS, T=80 ° C, V_{CC}=0,9 V)	284 mV

Le seuil de décision peut donc varier d'environ 60 mV entre le cas typique et l'un des deux cas extrêmes. Ceci peut poser des problèmes au niveau de la décision en elle-même – le seuil de décision peut ne pas être atteint dans le pire des cas, ou dans la réponse du fanal. En effet, si la valeur du seuil de décision change, la tension de sortie de l'élément WTA mettra plus ou moins de temps à l'atteindre selon le sens de variation du seuil. Le temps de réponse du fanal sera donc différent. Cette différence de temps de réaction se répercute également lors de l'assemblage des éléments entre eux. Un fanal complet conservera donc ce problème.

4.3.1.4 Effets des paramètres environnementaux sur le fanal en entier

Si les variations des paramètres environnementaux n'ont pas d'effet sur les valeurs des courants de sortie des synapses, le circuit WTA et le circuit de décision sont bien affectés, et le fonctionnement du fanal peut se trouver altéré. La valeur du seuil de décision doit en effet être choisie avec attention, afin de pouvoir discriminer un élément WTA vainqueur parmi tous les éléments WTA. Pour cela, le seuil de décision doit toujours se trouver entre les valeurs aux états "actif" et "inactif" de V_{COMP}, comme le montre la Figure 4.10. Il faut donc considérer 60 mV de marge sur le seuil de décision, dus aux variations de ce seuil lui-même, entre la valeur la plus haute de V_{COMP} à l'état "inactif" et la valeur la plus basse de V_{COMP} à l'état "actif". Le seuil de décision du buffer doit donc se trouver entre 107 mV et 375 mV.

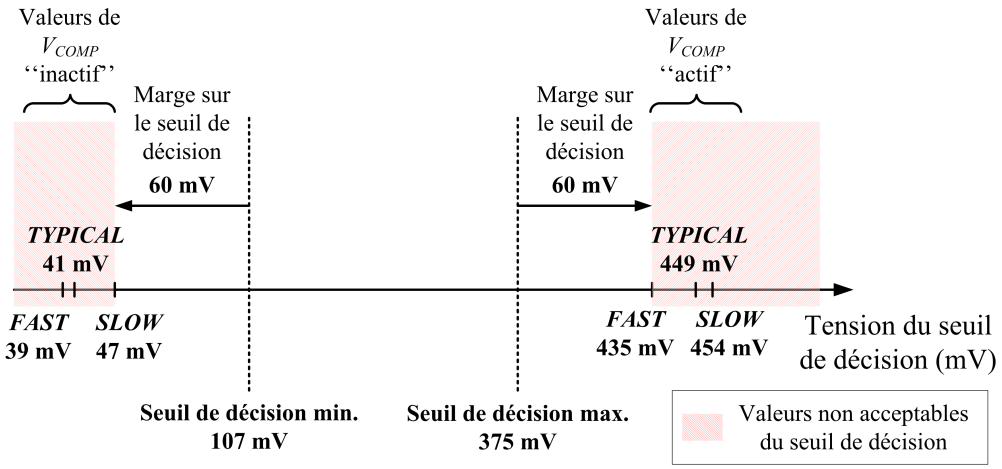

FIGURE 4.10 – Plage de valeur acceptable pour le seuil de décision, en prenant en compte tous les effets des variations des conditions environnementales dans le fanal.

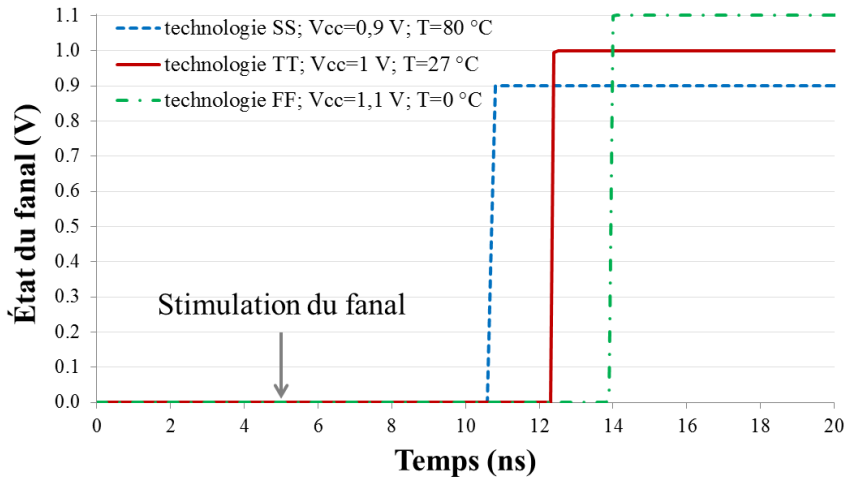

FIGURE 4.11 – Réponse du fanal à une unique stimulation, qui a lieu après 5 ns, dans différentes conditions environnementales.

Cependant, il reste un problème potentiel à considérer. Comme mentionné dans le Chapitre 3, le temps de réponse d'un fanal est le temps que met la tension V_{COMP} de l'élément WTA pour atteindre la valeur du seuil de décision. Si deux fanaux se trouvent dans des conditions environnementales différentes, leur seuil de décision sera différent et donc leur temps de réponse également.

Le temps de réponse à une stimulation d'un fanal dans différentes conditions environnementales est donné par la Figure 4.11. Dans cette simulation, on stimule une entrée du fanal à t=5 ns, pour V_{CC}=1 V, I_{UNIT}=300 nA et $V_{commande}$=350 mV. Le fanal répond à cette stimulation en 7,3 ns. Cette simulation est également conduite pour les conditions *FAST* et *SLOW*. On observe alors que le temps de réponse du fanal est modifié de 1,7 ns dans ces conditions, soit 23% du temps de

réponse nominal.

Un fanal peut donc propager une erreur dans le réseau plus rapidement qu'un autre va propager une donnée exacte. Ceci peut entraîner une convergence vers un message stocké différent de celui visé. Néanmoins, il n'est pas possible de conclure sur l'impact de ces variations sur un réseau complet, les erreurs éventuelles pouvant être corrigées par la redondance intrinsèque du réseau.

4.3.2 Impact sur la récupération des messages

Pour simuler l'impact des variations des paramètres environnementaux sur le réseau à cliques complet, on utilise le modèle comportemental du circuit avec *Simulink*®, décrit dans la Section 4.1.3. Pour une topologie donnée, ce modèle utilise des valeurs fixes pour les valeurs de V_{COMP} aux états "actif" et "inactif", ainsi que pour le seuil de décision. Ces valeurs étant tirées de simulations au niveau transistor effectuées grâce à *Spectre*®, le seuil de décision est toujours entre les valeurs de V_{COMP} aux états "actif" et "inactif".

Ces valeurs de tension fixes sont remplacées dans le modèle par des variables aléatoires gaussiennes. Ces variables aléatoires sont centrées sur les valeurs de chaque tension correspondant au cas *TYPICAL*. De plus, pour chaque variable aléatoire, la distribution de celle-ci est incluse à 99% dans la plage bornée par les cas extrêmes *FAST* et *SLOW*.

Pour une disposition des éléments du réseau comme celle décrite dans le Chapitre 2, Figure 2.2, les paramètres environnementaux peuvent être différents d'un cluster à un autre, qui peuvent être éloignés les uns des autres topologiquement parlant. En revanche, les fanaux à l'intérieur d'un cluster sont eux positionnés les uns à côté des autres. Nous pouvons donc raisonnablement supposer que les variations des paramètres environnementaux sont communes à un cluster. Nous

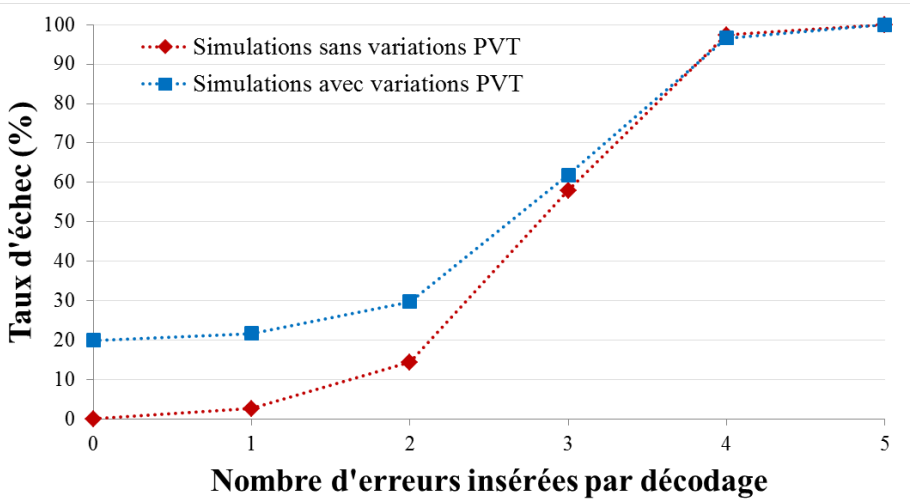

FIGURE 4.12 – Taux d'erreur du réseau en fonction du nombre d'erreurs introduites dans la stimulation. Les tests sont réalisés en simulation *Simulink*®, avec et sans variations des paramètres environnementaux.

les implémentons donc comme telles dans le modèle comportemental du circuit.

La nouvelle trace du réseau à cliques sujet à des variations des paramètres environnementaux est donnée sur la Figure 4.12. Les simulations sont effectuées pour V_{CC}=1 V, I_{UNIT}=300 nA et $V_{commande}$=350 mV. Cette trace est la moyenne de vingt calculs de traces, avec un nouveau tirage des variables aléatoires à chaque calcul.

Nous pouvons remarquer que même des cas où aucune erreur n'est insérée, le fait que des fanaux puissent réagir plus rapidement que d'autres peut générer des récupérations erronées. Ceci est accentué par la haute densité de notre réseau.

Garder un seuil de décision fixe comme dans les simulations de la Section 4.1.4 est donc important afin de ne pas dégrader les performances de récupération du réseau. Pour cela, un moyen de réaliser cette fonction au niveau circuit est de détecter les conditions environnementales dans lesquelles se trouve le cluster en question, et de les compenser artificiellement pour garder un seuil de décision fixe.

4.3.3 Circuit de compensation

La compensation des variations des paramètres environnementaux permet de garder un seuil de décision fixe, dont la valeur peut être ajustée de l'extérieur du réseau par une tension V_{ref}. Cette opération est réalisée au sein d'un cluster, et les résultats sont appliqués de manière identique à tous les fanaux du cluster.

Cette opération consiste à faire varier le courant circulant dans le buffer de décision de manière

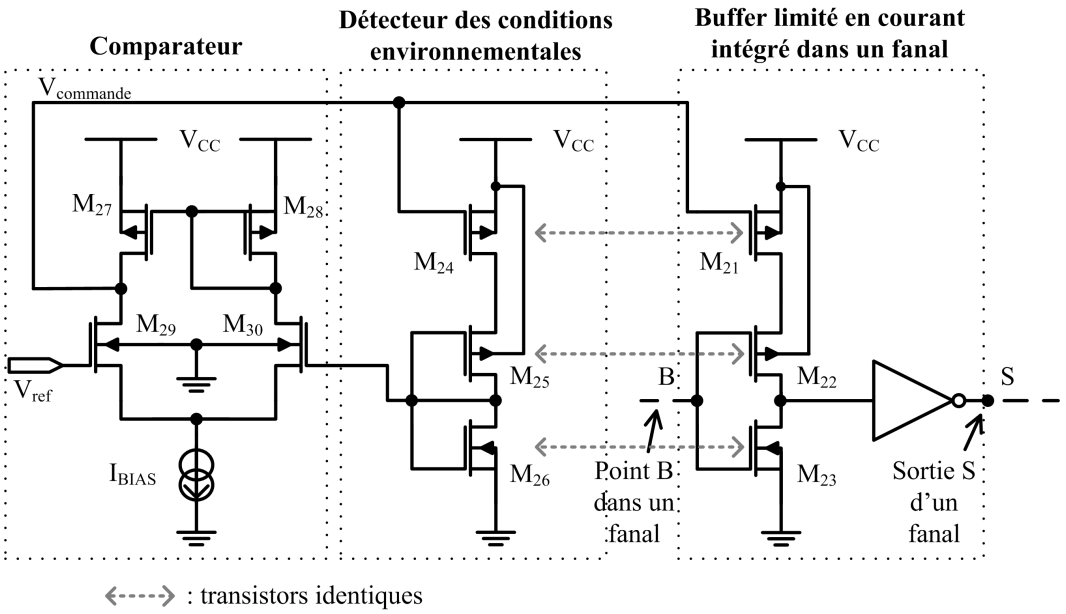

FIGURE 4.13 – Schéma électrique du circuit de compensation des variations des paramètres environnementaux.

à compenser les variations des paramètres environnementaux pour garder un seuil de décision fixe. Il faut donc détecter les conditions dans lesquelles se trouvent le cluster considéré, puis adapter la valeur du courant circulant dans le buffer de décision de tous les fanaux du cluster.

Le schéma électrique du circuit de compensation est donné en Figure 4.13. Afin de détecter les conditions dans lesquelles se trouvent le cluster considéré, un inverseur dont la sortie est connectée à l'entrée est placé dans le cluster, M_{24} à M_{26}. Cet inverseur est identique à l'inverseur limité en courant utilisé pour la décision. Le fait de reconnecter sa sortie à son entrée va stabiliser la tension à ce nœud au seuil de basculement de cet inverseur. La valeur de ce seuil est une image des conditions environnementales à compenser. Cette tension est alors comparée à la tension externe V_{ref} grâce à une paire différentielle formée des transistors M_{27} à M_{30}. Idéalement, V_{ref} est égal à la valeur du seuil de décision dans le cas *TYPICAL*. Le résultat de cette comparaison agit sur la tension $V_{commande}$ de l'inverseur limité en courant afin de stabiliser le circuit pour que son seuil de basculement se fixe à la tension V_{ref}. Cette compensation est appliquée à tous les fanaux du cluster en connectant le nœud $V_{commande}$ de chaque inverseur de décision au nœud $V_{commande}$ de l'inverseur de détection des conditions.

Le Tableau 4.5 donne les nouvelles valeurs du seuil de décision, avec comme objectif un seuil de décision V_{ref} de 350 mV. Les résultats sont issus de simulations effectués avec *Spectre*®.

TABLEAU 4.5 – Valeurs du seuil de décision en fonction des conditions environnementales, en incluant le dispositif de compensation.

Valeur des paramètres environnementaux	Seuil de décision du buffer sans compensation	Seuil de décision du buffer avec compensation
Corner *"TYPICAL"* (technologie TT, T=27 ° C, V_{CC}=1 V)	350 mV	350 mV
Corner *"FAST"* (technologie FF, T=0 ° C, V_{CC}=1,1 V)	407 mV	353 mV
Corner *"SLOW"* (technologie SS, T=80 ° C, V_{CC}=0,9 V)	284 mV	345 mV

Les valeurs du seuil de décision sont alors stabilisées autour de la tension V_{ref}. Leurs variations sont réduites de 76 mV sans compensation, et à 5 mV avec. Pour vérifier si le temps de réponse d'un fanal ne varie plus selon les conditions environnementales, la simulation de la Section 4.3.1.4 est reconduite. La réponse d'un fanal à cette stimulation, dans différentes conditions, est représentée sur la Figure 4.14.

La réponse d'un fanal est donc harmonisée quelles que soient les conditions environnementales. Le seuil de décision peut donc être considéré comme fixe dans le circuit, et les performances du circuit sont identiques à celles du modèle idéal, en considérant uniquement ces imperfections. Ces

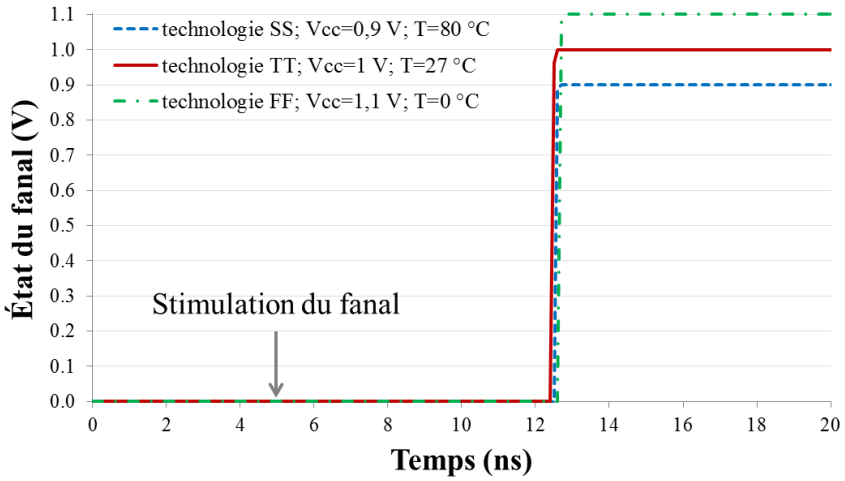

FIGURE 4.14 – Réponse d'un fanal avec cellule de compensation des paramètres environnementaux à une unique stimulation, qui a lieu après 5 ns, dans différentes conditions environnementales.

performances de récupération d'information correspondent à la courbe sans variations PVT dans la Figure 4.12.

Cette section a montré les problèmes apportés par les variations des paramètres environnementaux. Une fois identifiés, des moyens de compensation ont été mis en place afin de retrouver les performances idéales.

Conclusion

Ce chapitre a fixé la taille du réseau qui sera intégré sur ASIC, un réseau de cinq clusters de six fanaux chacun, avec dix mots stockés dans son dictionnaire. Il a ensuite défini les performances selon lesquelles le circuit du réseau à cliques sera évalué : pouvoir de récupération d'information, temps de réponse et puissance consommée.

Un protocole de simulation a alors été mis en place afin d'évaluer les performances du réseau à cliques étudié. Pour cela, un modèle comportemental a été mis en œuvre grâce à *Simulink*®.

Après avoir évalué les performances du circuit dans les conditions idéales, l'impact d'imperfections inhérentes à l'intégration sur ASIC sur le circuit a été étudié. Les performances globales du circuit se trouvant dégradées, des moyens de compensation et de minimisation de ces effets parasites ont été intégrés au circuit du réseau à cliques.

Le circuit est alors robuste, et prêt à son intégration sur puce. Le chapitre suivant présente le circuit envoyé en fabrication, ainsi que le banc de test utilisé pour les mesures de la puce obtenue. Enfin, les résultats de mesures sur la puce reçue sont exposés à la fin du prochain chapitre.

Chapitre 5

Test et mesures d'un réseau à cliques intégré sur puce

Introduction

Le chapitre précédent a montré les performances que l'on peut espérer du circuit en termes de pouvoir de récupération d'information, en simulation. Il a aussi été montré que les défauts d'appariement des transistors, ainsi que les variations des paramètres environnementaux, peuvent nuire au bon fonctionnement du circuit. Des dispositifs de compensation ont donc été mis en place afin de rendre le circuit robuste dans une intégration sur un ASIC. Ce chapitre décrit la composition et le test d'un ASIC de 1 mm^2 intégrant le circuit d'un réseau de neurones à cliques présenté dans les chapitres précédents. Le design kit ST CMOS 65 nm est utilisé pour créer les masques de ce circuit. Ce dernier intègre un réseau de neurones à cliques composé de cinq clusters de six fanaux, dont les performances ont été simulées dans le Chapitre 4. Il intègre aussi les dispositifs de compensation des variations des conditions environnementales.

La Section 5.1 présente tout d'abord les objectifs des mesures, et les tests mis en œuvre pour obtenir les résultats désirés.

La Section 5.2 de ce chapitre présente les éléments utilisés dans le test de l'ASIC. Elle donne notamment la composition du circuit intégré sur ASIC. Des modules sont de plus intégrés afin de tester les fonctions élémentaires du réseau de neurones à cliques, comme le circuit WTA. Cette section présente enfin le matériel utilisé pour polariser, stimuler et récupérer les résultats de l'ASIC.

La Section 5.3 donne ensuite les résultats des tests effectués sur l'ASIC. Ces résultats sont comparés aux résultats théoriques donnés par la simulation du même réseau. Enfin, le fonctionnement de l'ASIC est testé dans des conditions environnementales différentes du cas typique, afin de tester sa robustesse.

Enfin, la Section 5.4 compare les performances du circuit produit à celles d'un équivalent numérique, en termes de surface de silicium occupée et de latence.

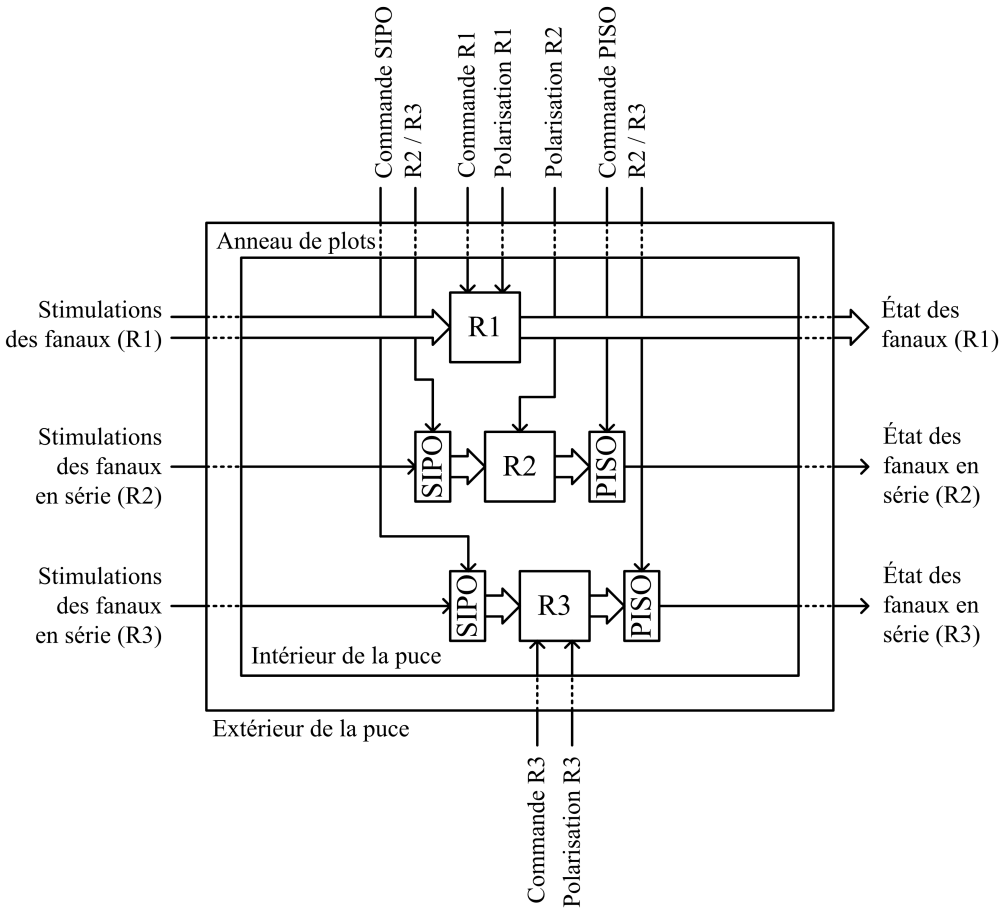

FIGURE 5.1 – Schéma structurel du circuit intégré sur la puce. Trois réseaux $R1$, $R2$ et $R3$ servent à tester différents aspects du circuit.

5.1 Objectifs des mesures

Le principal objectif de la fabrication d'un ASIC intégrant un réseau de neurones à cliques est de vérifier la fonctionnalité des éléments de circuit décrits dans le Chapitre 3. Si cela s'avère correct, il convient alors de comparer ses performances avec les simulations comportementales : temps de réponse d'un fanal, temps de convergence d'un réseau complet, pouvoir de récupération d'information du réseau, consommation du circuit et sa robustesse en regard des variations des conditions environnementales et du défaut d'appariement des transistors.

Le schéma structurel du circuit intégré est donné dans la Figure 5.1. Chaque partie de ce circuit intégré et les mesures envisagées pour chaque partie sont décrites dans la suite de cette section.

Avant de vérifier le comportement d'un réseau complet, il faut valider le comportement du circuit WTA dans un cluster de fanaux. Nous avons donc conçu un réseau, appelé réseau $R1$ dans la suite du document, intégrant un unique cluster de six fanaux. La longueur du cluster est définie afin de garder les éléments passifs parasites intrinsèques au cluster identiques à ceux de notre réseau

complet cible, c'est-à-dire cinq clusters de six fanaux. Dans ce cluster, les stimulations des fanaux sont contrôlées depuis l'extérieur de la puce, afin de pouvoir tester les cas de supériorité d'un fanal et les cas d'égalité entre fanaux. Les sorties des fanaux sont observables depuis l'extérieur de la puce, pour montrer la fonctionnalité du circuit WTA et mesurer aussi le temps de réponse d'un fanal à une stimulation. Une entrée de polarisation additionnelle est ajoutée pour maîtriser depuis l'extérieur le seuil du buffer de décision d'un fanal. Cela permettra de valider le fonctionnement du circuit de compensation des variations des paramètres environnementaux.

Pour valider le fonctionnement d'un réseau complet, un réseau de cinq clusters de six fanaux, appelé $R2$, est conçu. Il représente le stockage de mots de cinq lettres, composés avec un alphabet limité. Afin de simplifier au maximum les échanges entre fanaux, nous choisissons de les relier par des pistes métalliques continues. Cela fixe le stockage des informations, mais assure une transmission rapide des données. Nous souhaitons stocker dans ce réseau un maximum d'informations possible sans perdre de performance de récupération d'information. Dix mots sont donc stockés dans le dictionnaire, et la répartition des connexions n'est pas uniforme dans tout le réseau. Les entrées de stimulation et les sorties du réseau doivent être accessibles, afin de réaliser les tests de performance simulés dans le Chapitre 4.1.1.1. Cependant, cela nécessite soixante plots d'entrées/sorties pour tout rendre accessible. Or, le nombre de plots disponibles sur une puce de 1 mm^2 est, au maximum, de 44, en comptant les plots des entrées d'alimentation et de polarisation. Deux registres PISO (Parallel In - Serial Out) et SIPO (Serial In - Parallel Out) sont donc mis en place afin de d'accéder aux entrées de stimulation et aux sorties des fanaux en série. Les tests réalisés en simulation dans le Chapitre 4.1.1.1 permettront alors de mettre en évidence le pouvoir de correction du réseau, son temps de convergence et la consommation du circuit. En faisant varier la tension d'alimentation et la température du banc de test, nous pourrons aussi montrer l'intérêt de la cellule de compensation des variations des paramètres environnementaux.

Enfin, il est intéressant de pouvoir décomposer les échanges d'informations pour étudier des cas où la récupération d'information ne se passe pas de la même façon dans les mesures et en simulation. Pour cela, nous avons conçu un autre réseau de cinq clusters de six fanaux, appelé $R3$. La différence avec le réseau $R2$ est la présence d'un circuit numérique permettant ou non la propagation de l'état des fanaux vers les autres. Comme dans le réseau $R2$, des registres SIPO et PISO doivent être ajoutés en entrée et en sortie du réseau pour accéder aux états des fanaux.

Le Tableau 5.1 récapitule les différentes caractéristiques du réseau à mesurer, sur quelle partie du circuit, et les valeurs des métriques obtenues en simulation dans le Chapitre 4.

TABLEAU 5.1 – Rappel des grandeurs à mesurer sur le circuit intégré.

Grandeur à mesurer	Résultat obtenu en simulation	Réseau à utiliser pour la mesure
Temps de réponse d'un fanal	7,3 ns	$R1$
Temps de convergence du réseau à cliques	21 ns	$R2$
Pouvoir de récupération d'information du réseau à cliques	voir Chapitre 4.1.4, Figure 4.3	$R2$
Consommation du réseau à cliques	3,1 μW par fanal inactif et 5,8 μW par fanal actif	$R2$
Robustesse aux conditions environnementales	Pas de perte de performances en conditions *TYPICAL*, *FAST* ou *SLOW*	$R2$
Décomposition de la récupération d'un message		$R3$

5.2 Présentation des éléments de test

5.2.1 Présentation des circuits dans la puce

5.2.1.1 Réseau $R1$: Cluster de test

Le fonctionnement de ce réseau doit permettre de vérifier :
- le bon fonctionnement du circuit WTA dans plusieurs cas de stimulations ;
- le temps de réponse d'un fanal à une stimulation ;
- le fonctionnement de la cellule de compensation des variations des paramètres environnementaux ;
- si ce dernier n'est pas bon, donner la possibilité de le contourner.

Le schéma structurel de ce réseau est donné sur la Figure 5.2. Le réseau $R3$ est alimenté par $AVCC2$, et est composé d'un seul cluster de six fanaux, du fanal #1 au fanal #6. Chaque fanal est de plus connecté à deux synapses, de façon à ce que les capacités parasites des synapses (cf. Chapitre 3) soient identiques pour chaque fanal.

Trois entrées de stimulation sont mises en place et commandables de façon externe dans ce réseau : deux sur le fanal #1 ($Stim\_F1\_1\_R1$ et $Stim\_F1\_2\_R1$) et une sur le fanal #2 ($Stim\_F2\_R1$). De cette manière, on peut vérifier les cas de supériorité d'un fanal par rapport à un autre, ainsi que les cas d'égalité entre fanaux grâce à des mesures en stimulant ces entrées. Pour

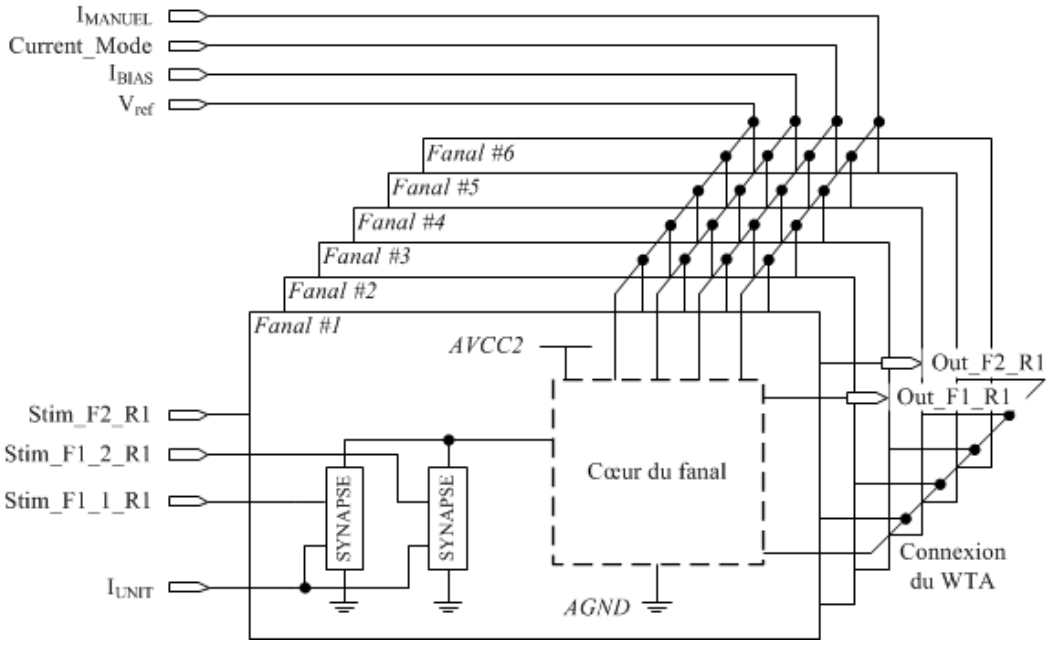

FIGURE 5.2 – Schéma structurel du réseau $R1$.

ceci, les états des fanaux #1 et #2 sont sortis de la puce sous la forme des signaux Out_F1_R1 et Out_F2_R1.

De plus, un module de compensation des paramètres environnementaux (cf. Chapitre 4) est mis en place dans ce réseau. Il est polarisé par les signaux I_{BIAS} et V_{ref}. Un réglage alternatif du seuil de décision par l'entrée I_{MANUEL} dans les fanaux est aussi intégré à ce réseau. Le signal de commande $Current\_Mode$ permet d'activer l'un ou l'autre des dispositifs de réglage du seuil de décision.

Le layout de ce réseau est montré sur la Figure 5.3. La surface totale du réseau $R1$ est de 780 μm^2.

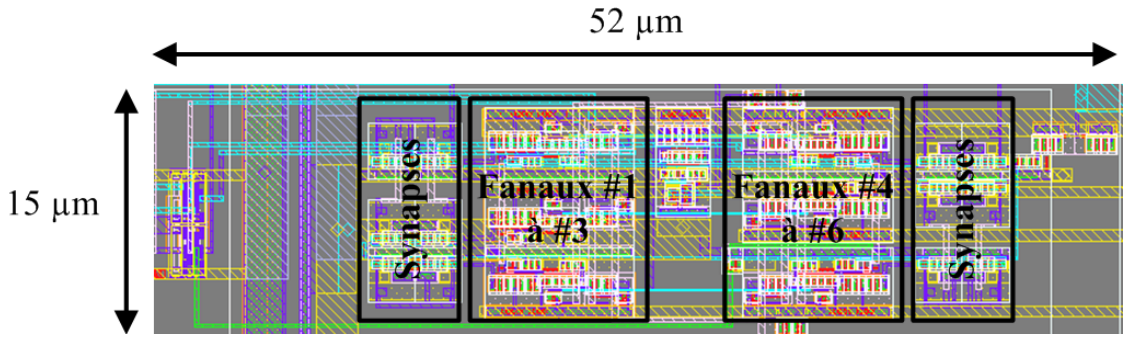

FIGURE 5.3 – Layout du réseau $R1$ complet. Sa position est indiquée sur la photographie du circuit intégré dans la Section 5.2.2.

5.2.1.2 Réseau $R2$: Réseau “continu”

L'objectif de ce réseau est de vérifier les performances de récupération d'information du réseau à cliques étudié dans le Chapitre 4. Plus précisément, il s'agit de vérifier :

- le bon fonctionnement d'un réseau à cliques complet, l'allure de son pouvoir de récupération d'information ;
- le temps de convergence et la consommation du réseau ;
- la robustesse du réseau aux variations des paramètres environnementaux.

Pour cela, le réseau $R2$ utilise la même application que dans le Chapitre 4. Il s'agit de retrouver des mots de cinq lettres dans un dictionnaire. Ces mots sont réalisés à partir d'un alphabet limité aux lettres A, E, I, R, S et T. Le réseau $R2$ est donc composé de cinq clusters de six fanaux chacun. Le dictionnaire est composé des dix mots présents dans l'application du Chapitre 4, et est rappelé dans le Tableau 5.2.

TABLEAU 5.2 – Dictionnaire des mots stockés dans le réseau $R2$.

STARS	SITAR
ARTIS	TARSI
RAISE	RESIT
TEARS	EARST
TASER	ISETA

Dans le réseau $R2$, il n'est pas possible de sortir tous les signaux d'état des fanaux de la puce en parallèle, ni d'entrer tous les stimuli. Pour pallier ce problème, les signaux de stimulation et les signaux de sortie sont sérialisés. Des registres SIPO et PISO sont insérés en amont et en aval du

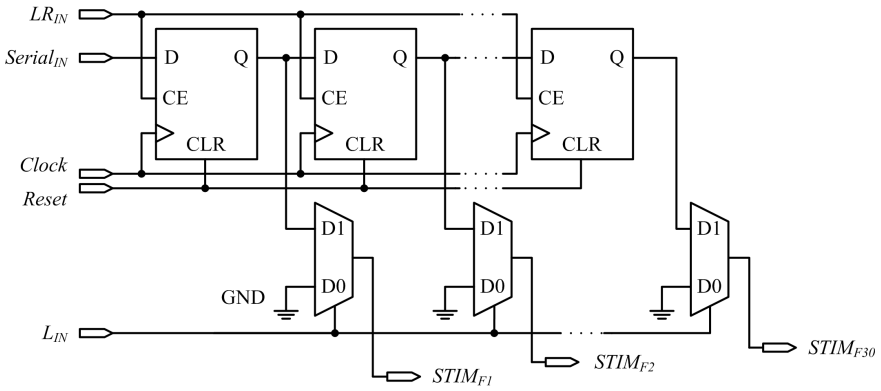

FIGURE 5.4 – Schéma du registre SIPO.

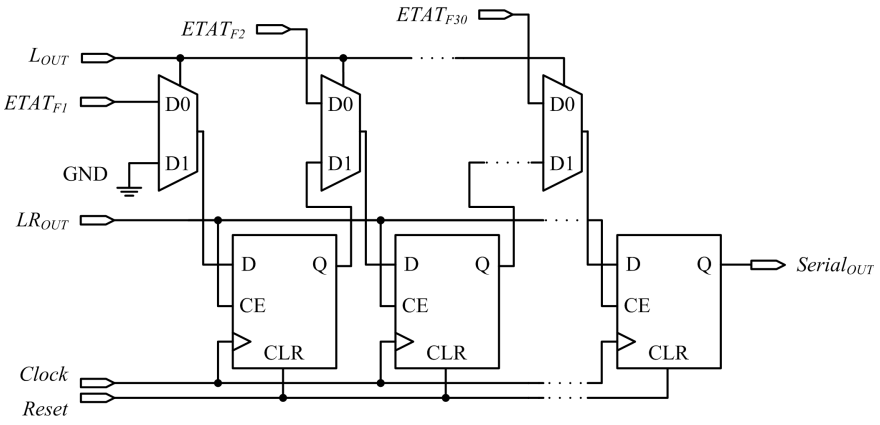

FIGURE 5.5 – Schéma du registre PISO.

réseau, Figures 5.4 et 5.5. Les signaux de commande (*Clock*, *Reset*, LR_{IN}, L_{IN}, LR_{OUT}, L_{OUT}, $Serial_{IN}$) sont générés hors de la puce et sont fournis directement aux registres. Dans le registre SIPO, le signal LR_{IN}, actif à '1', permet à l'entrée $Serial_{IN}$ d'avancer bit à bit dans le registre. Le message d'entrée est donc dans le registre au bout de 30 cycles d'horloge. Le signal L_{IN} peut alors envoyer tous les stimuli en même temps dans le réseau. Dans le registre PISO, le signal L_{OUT} à '0' permet de charger les sorties dans le registre, et à '1' de connecter les éléments du registre entre eux. Le signal LR_{OUT} permet de faire avancer les données à l'extérieur de la puce sous la forme du signal $Serial_{OUT}$.

Le réseau $R2$ est alimenté par $AVCC1$. Le schéma structurel du réseau $R2$ est donné sur la Figure 5.6. Chaque fanal peut être stimulé une fois depuis l'extérieur grâce au signal $Serial_{IN}$ et au registre SIPO. Chaque fanal est connecté à treize synapses. Certaines synapses ne sont donc pas connectées, car le stockage n'est pas uniforme dans le réseau. Cependant, il est important de les laisser afin que tous les fanaux aient le même temps de réponse, comme expliqué dans le Chapitre 3, Section 3.5.1. Les sorties des fanaux sont connectées aux synapses d'autres fanaux par des pistes métalliques directes. Les lignes verticales sont connectées au niveau de métal *M1*, tandis que les lignes horizontales sont connectées au niveau de métal *M2*. De plus, les pistes permettant de transmettre des données numériques ne passent pas au dessus des cellules analogiques, car elles génèrent des phénomènes parasites nuisant au bon fonctionnement des cellules analogiques.

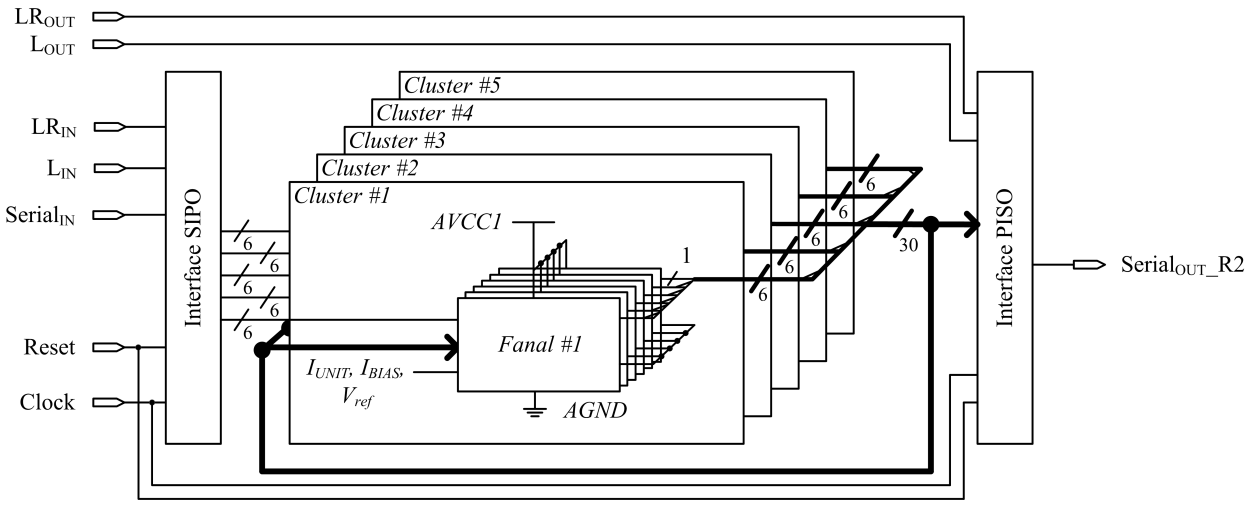

FIGURE 5.6 – Schéma structurel du réseau $R2$.

La Figure 5.7 montre le déroulement de la récupération d'un message. Une fois que le signal *Reset* (actif à '0') est inactif, le signal LR_{IN} est activé et le mot de stimulation contenu dans le signal $Serial_{IN}$ est présenté en entrée du réseau. Après trente cycles d'horloge, on active le signal L_{IN} pour stimuler le réseau. On laisse ensuite le réseau converger vers une solution en laissant L_{IN} actif. Une fois cette phase finie, le signal LR_{OUT} est activé afin d'enregistrer les sorties du réseau. Le cycle d'après, L_{OUT} est activé à son tour et le résultat est transféré sur le signal $Serial_{OUT}\_R2$, durant trente cycles d'horloge. Le signal *Reset* peut alors être activé pour se préparer à la prochaine récupération de message.

Le pouvoir de récupération d'information du réseau est mesuré en effectuant ce test pour les 16 807 stimuli différents (cf. Chapitre 4, Section 4.1.4). Le résultat de chaque récupération de message est comparé au mot du dictionnaire correspondant. Ce test nécessite donc une partie contrôle pour cadencer chaque récupération, récupérer les données de sortie et les comparer aux mots du dictionnaire, et démarrer une nouvelle récupération de message. De plus, il est possible de modifier le temps laissé au réseau pour converger en changeant le nombre de cycles d'horloge où L_{IN} est actif. En diminuant cette durée, il arrive un moment où le réseau n'a pas le temps de converger dans tous les cas. Cela permet d'avoir une mesure du temps de convergence du réseau, c'est-à-dire la durée à partir de laquelle on aperçoit un affaiblissement du pouvoir de récupération d'information. La consommation du cœur analogique du réseau est mesurée en relevant l'intensité du courant provenant de l'alimentation $AVCC1$. Enfin, pour vérifier la robustesse du réseau aux variations des paramètres environnementaux, il suffit de faire varier la valeur de la tension d'alimentation $AVCC1$ de plus ou moins 10%. Une étuve peut de plus être utilisée pour changer la température à laquelle fonctionne le circuit.

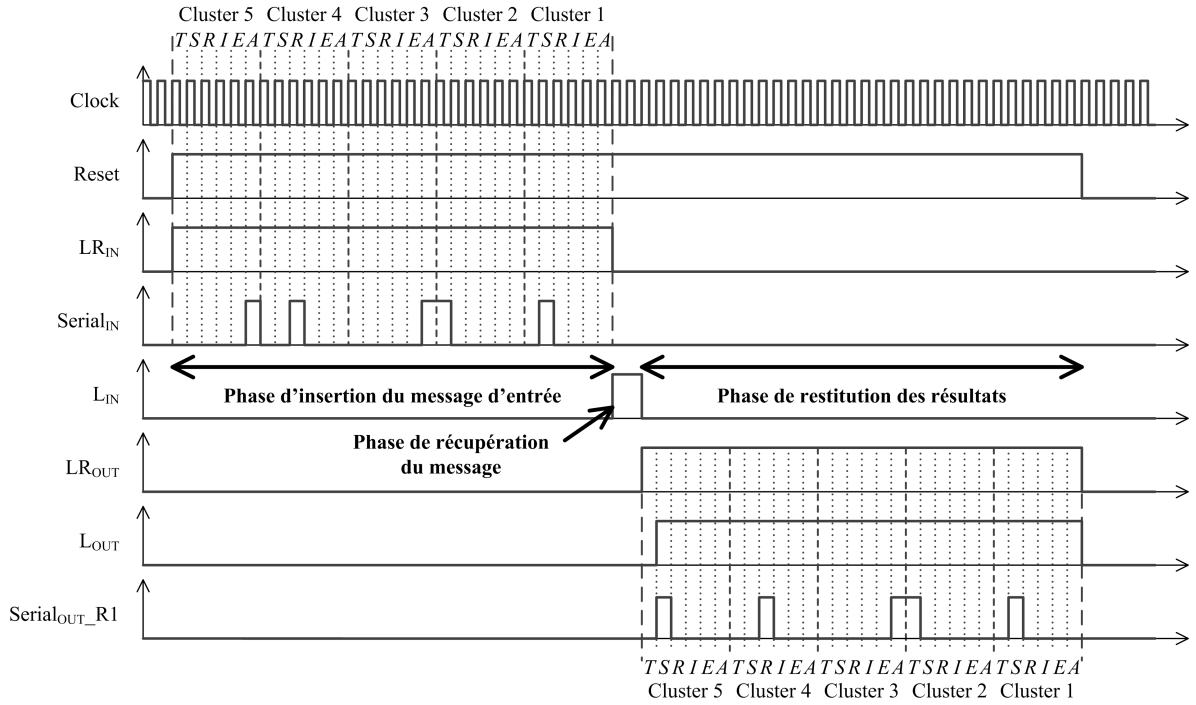

FIGURE 5.7 – Chronogramme décrivant le déroulement de la récupération d'un message dans le réseau $R2$.

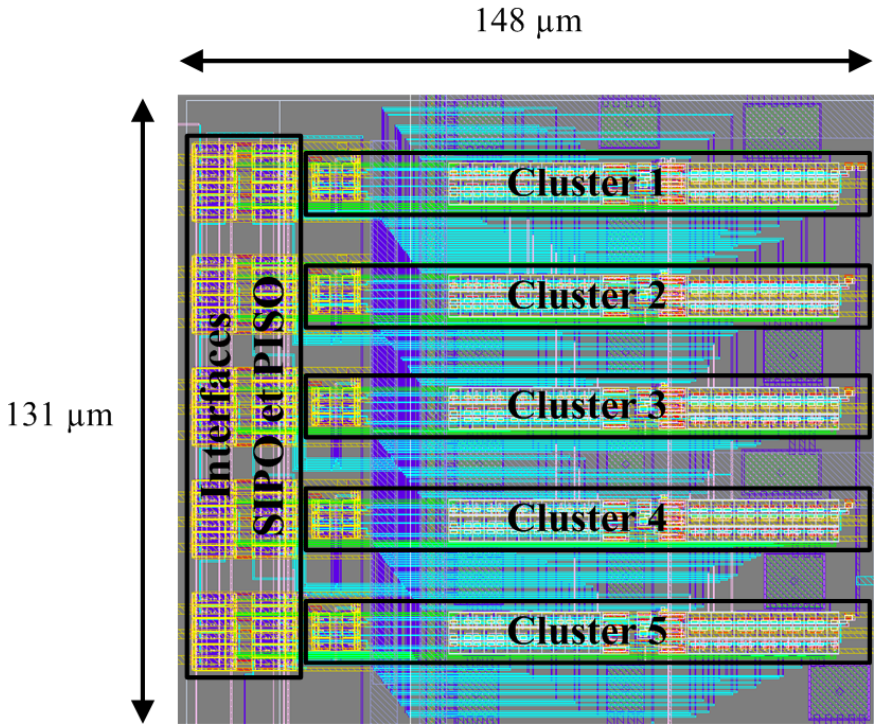

FIGURE 5.8 – Layout du réseau $R2$ complet. Sa position est indiquée sur la photographie du circuit intégré dans la Section 5.2.2.

Le layout de ce réseau est montré sur la Figure 5.8. La surface totale du réseau $R2$ est de 19 388 μm^2.

5.2.1.3 Réseau $R3$: Réseau "bus commun"

L'objectif de ce réseau est le suivant : pouvoir décomposer les itérations d'une récupération de message afin d'observer d'éventuelles erreurs non prévues durant son déroulement. La structure de ce réseau est exactement la même que celle du réseau $R2$ (cinq clusters de six fanaux, dictionnaire identique). La différence avec le réseau $R2$ est que la communication entre les fanaux se fait grâce à un bus de six signaux commun à tous les clusters. Chacun à leur tour, les clusters prennent possession du bus et envoient leurs informations aux autres fanaux. L'état de chaque fanal est mémorisé et est envoyé ou non sur le bus. En amont des synapses, des mémoires sont placées afin de "filtrer" les informations ne provenant que des clusters qui les concernent. Elles permettent également de retenir l'information sans la propager dans le réseau tant que tous les clusters n'ont pas propagé leurs informations.

Cette structure a pour objectif de diviser la récupération d'un message en itérations. De cette façon, il est plus facile de vérifier ce qui se passe effectivement durant la récupération et d'analyser les échanges entre fanaux. Une autre utilisation possible de ce réseau est de connecter deux puces ensemble via le bus commun. De cette façon, il est possible de remplacer un cluster par son équivalent dans l'autre puce. Avec un réseau reconfigurable, il serait même possible de rajouter dynamiquement des clusters à un réseau de cette façon.

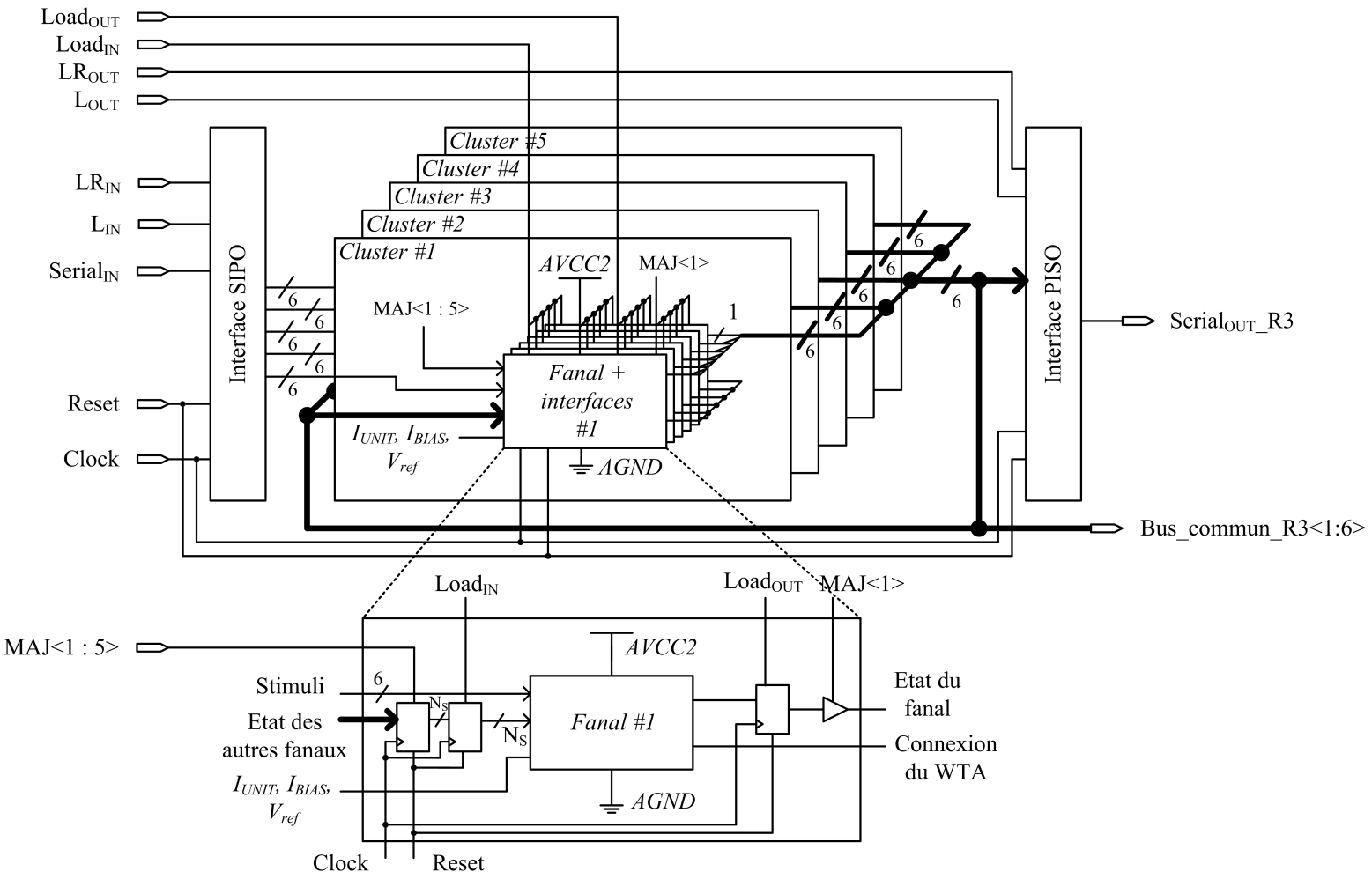

FIGURE 5.9 – Schéma structurel du réseau $R3$. Les étages de mémoires sont uniquement représentés sur le cluster #1, mais sont aussi présents dans les autres clusters.

Le réseau $R3$ est alimenté par $AVCC2$. La Figure 5.9 montre le schéma structurel de ce réseau. La synchronisation des échanges entre les clusters doit être gérée par du matériel supplémentaire car ces échanges ne sont plus continus, mais séquentiels par cluster. Des points mémoires commandés par le signal $Load_{OUT}$ sont placés en sortie des fanaux pour échantillonner l'information après chaque itération. Des buffers trois états, commandés par les signaux $MAJ < 1 : 5 >$, permettent ou non l'accès au bus par chaque cluster successivement. Un étage de points mémoire placés en amont des synapses, commandés par les signaux $MAJ < 1 : 5 >$, permet de filtrer l'information qui arrive à chaque fanal en fonction du cluster qui accède au bus de données principal. Enfin, afin d'actualiser l'état des fanaux dans tout le réseau simultanément, un deuxième étage de points mémoire est placé en série après le premier, et est commandé par le signal $Load_{IN}$.

Le déroulement de la récupération d'un message est donné sur la Figure 5.10. Le chargement des signaux de stimulation ne change pas par rapport au réseau $R2$. Une fois que le réseau a convergé, le signal $Load_{OUT}$ est activé afin de mémoriser les états de chaque fanal. Chaque cluster va ensuite prendre possession du bus commun en activant les signaux $MAJ < 1 : 5 >$ les uns après les autres.

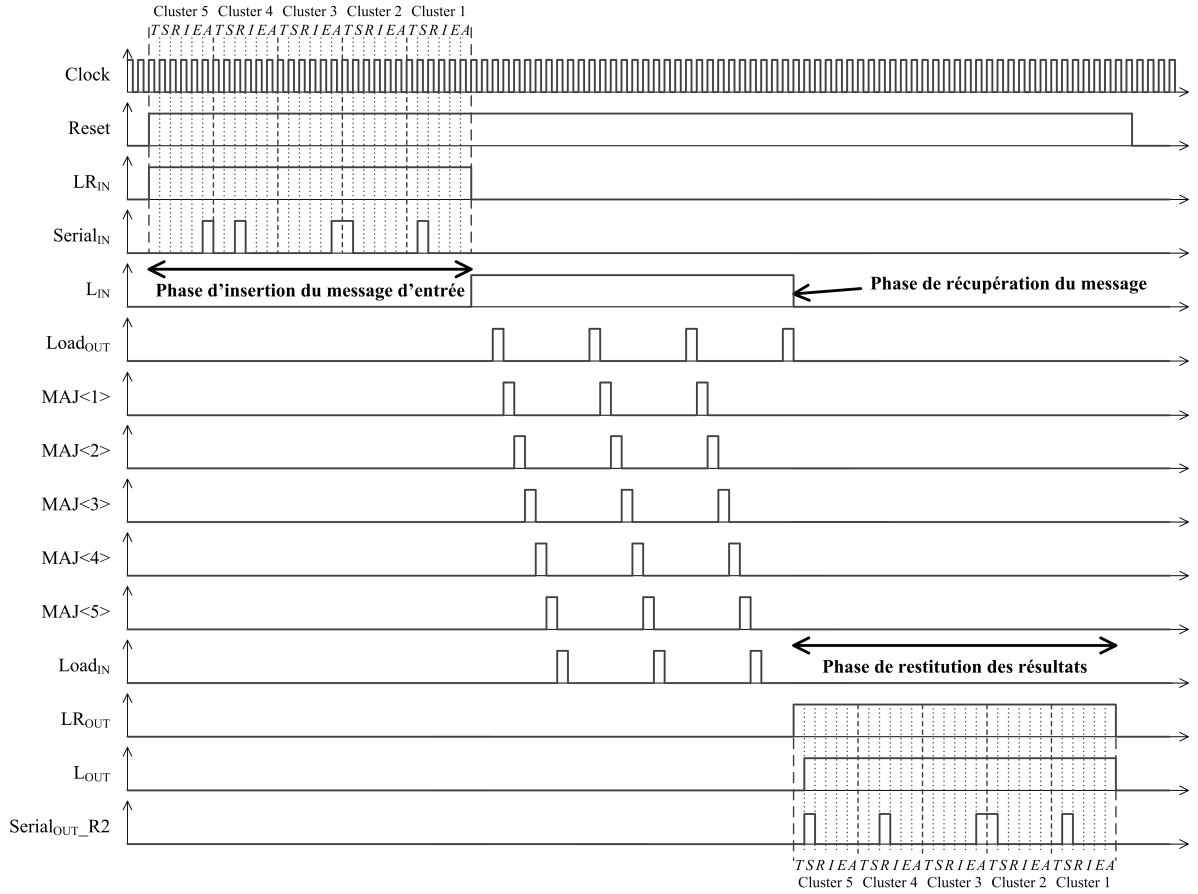

FIGURE 5.10 – Chronogramme décrivant le déroulement de la récupération d'un message dans le réseau $R3$.

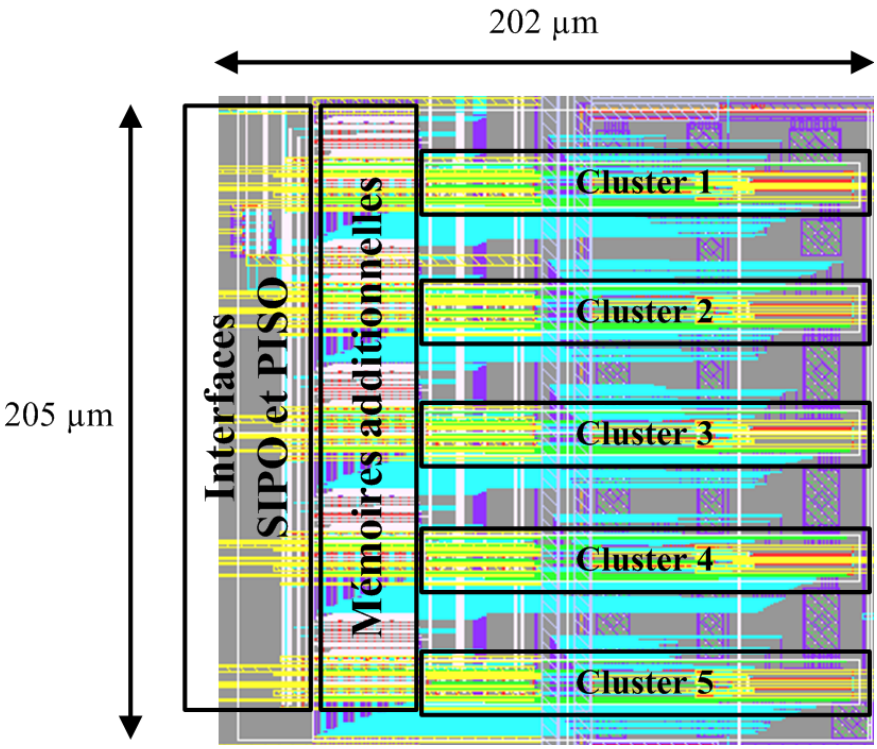

FIGURE 5.11 – Layout du réseau $R3$ complet. Sa position est indiquée sur la photographie du circuit intégré dans la Section 5.2.2.

Ceux-ci contrôlent aussi la mémorisation des informations en amont des synapses pour chaque cluster. Le signal $Load_{IN}$ s'active enfin pour libérer ces informations dans le réseau en même temps. Un cycle de convergence est alors effectué, et ce jusqu'à arriver à un nombre d'itérations suffisant pour arriver à un résultat final. D'après [GB11], quatre itérations sont suffisantes pour arriver à un résultat stable dans tous les cas. La restitution du résultat se déroule de la même manière que dans le réseau $R2$, à ceci près que chaque cluster doit charger son information après que le précédent ait restitué la sienne (fait quand L_{OUT} passe à '0').

Une mesure de la consommation de ce réseau n'est pas possible, car le réseau $R1$ est également alimenté par $AVCC2$. Il n'est alors pas possible de discriminer leurs consommations.

Le layout de ce réseau est montré sur la Figure 5.11. La surface totale du réseau $R3$ est de 41 820 μm^2.

5.2.2 Présentation du circuit intégré

Le circuit intégrant le réseau de neurones à cliques est une puce carrée de 1,026 mm de côté, soit 1,05 mm^2. 25 exemplaires de cette puce (de la puce #1 à la puce #25) ont été reçus, ce qui permet une étude du fonctionnement du réseau sur plusieurs échantillons, notamment en ce qui concerne les défauts d'appariement des transistors ou les paramètres environnementaux. La disposition des

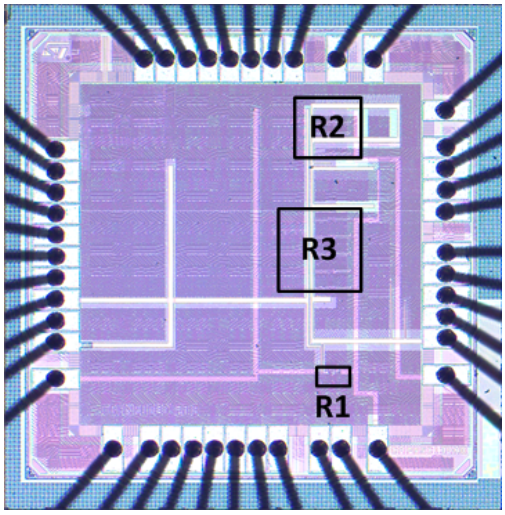

FIGURE 5.12 – Micro-photographie du circuit intégré indiquant la position des réseaux $R1$, $R2$ et $R3$.

réseaux $R1$, $R2$ et $R3$ sur la puce est donnée sur la Figure 5.12.

Des techniques de conception de masques ont été appliquées afin de protéger le circuit de perturbations éventuelles. Ainsi, les alimentations analogiques et numériques sont séparées et le substrat de la puce est divisé en deux parties isolées électriquement. Aucune piste transportant des données numériques ne passe au dessus d'une cellule analogique pour ne pas créer d'interférence. Enfin, des anneaux de garde de types N et P entourent les parties analogiques à l'intérieur de la puce, permettant de filtrer des porteurs de charge parasites provenant des parties numériques.

Afin de réduire encore le nombre de plots d'entrées/sorties, les signaux de commande des différents réseaux sont mutualisés si possible. La disposition des entrées/sorties de la puce est montrée sur la Figure 5.13. L'ensemble des plots de la puce est divisé en deux anneaux distincts distribuant l'énergie aux plots et au cœur de la puce, un analogique et un numérique. Chaque anneau a sa propre alimentation ($AVCCE$ et $DVCCE$) et masse ($AGNDE$ et $DGNDE$), et distribue les alimentations et masses des parties analogiques et numériques des circuits ($AVCC1$, $AVCC2$ et $DVCC$, $AGND$ et $DGND$). Les plots sont soit des plots analogiques, soit des plots numériques. Les plots analogiques sont de simples liaisons métalliques, tandis que les plots numériques font l'interface entre les domaines de tension interne et externe à la puce, pour les signaux numériques. Tous ces plots sont standards et sont fournis dans le kit.

5.2.3 Présentation du banc de test

La Figure 5.14 montre le banc de test dans son ensemble. Les entrées de la puce sont générées de trois manières différentes. Tout d'abord, les signaux d'alimentation sont produits à partir de générateurs continus *Metrix* AX503 permettant une précision au dixième de volt. Les alimentations des anneaux (analogique et numérique) sont fixées à 2,5 V, tandis que les alimentations des

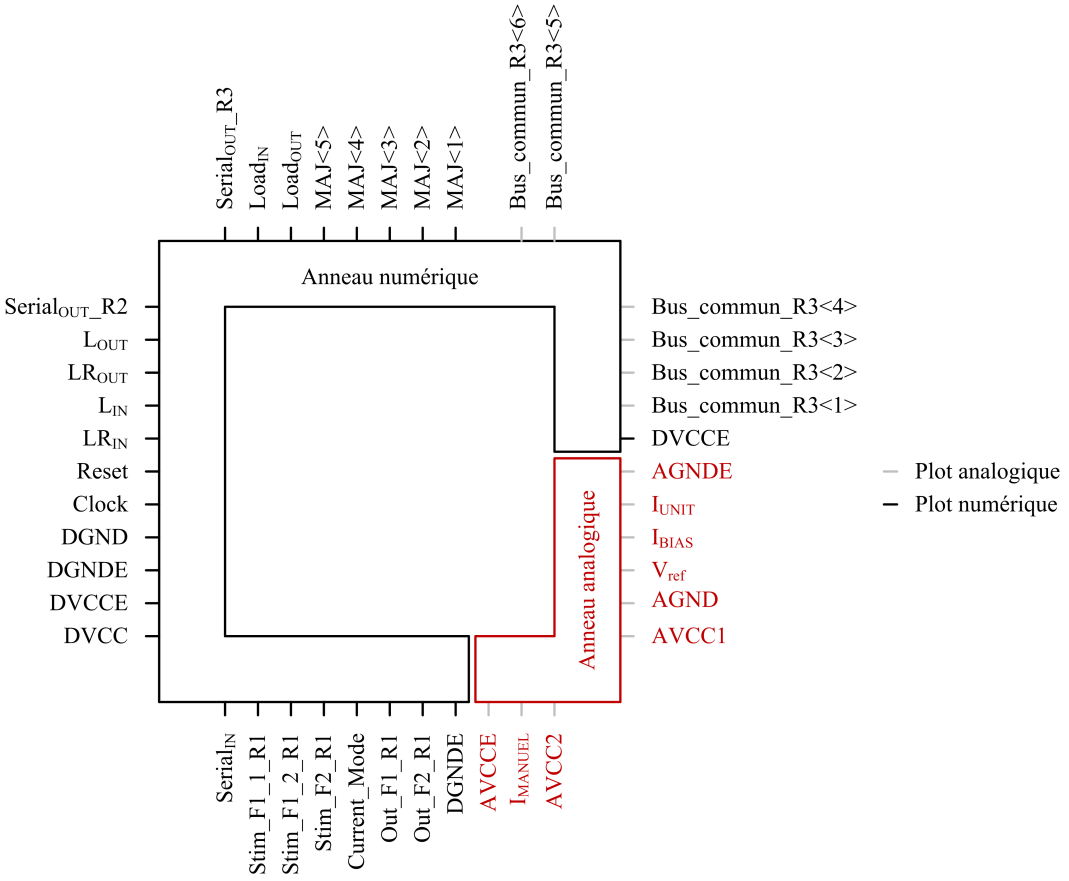

FIGURE 5.13 – Disposition des plots du circuit intégré.

cœurs (analogique et numérique) sont fixées à 1 V. En tout, deux générateurs sont utilisés pour l'alimentation du circuit, et deux canaux par générateur.

Un autre canal de générateur est utilisé pour produire V_{ref}. Les courants de polarisation I_{UNIT}, I_{BIAS} et I_{MANUEL} sont générés à partir de résistances. Ces dernières sont choisies variables (de 1 $M\Omega$ à 5 $M\Omega$), afin de pouvoir les ajuster manuellement. Des multimètres *Agilent* 34401A et *HP* 3478A sont aussi utilisés pour vérifier la valeur des courants de polarisation.

Les signaux de commande numériques sont générés par une carte FPGA *Atlys*® (FPGA Spartan-6) de *Digilent*, Figure 5.15. Cette carte permet aussi de choisir un mode de fonctionnement : test du réseau $R1$, test du réseau $R2$ ou test du réseau $R3$ séparément. Le signal d'horloge de la carte est généré par un générateur de signaux arbitraires *Agilent* 33120A.

Les sorties des réseaux sont visualisées grâce à un oscilloscope *LeCroy WaveSurfer*® 104MXs-A. De plus, ces sorties sont aussi récupérées par le FPGA et envoyées vers un tableur afin d'analyser les résultats.

Enfin, pour faire l'interface entre le matériel de test et le circuit intégré, une carte de test est réalisée, Figure 5.16. Elle permet aussi d'intégrer un étage de filtrage pour tous les canaux d'alimentation. Les résistances permettant la génération des courants de polarisation sont intégrées

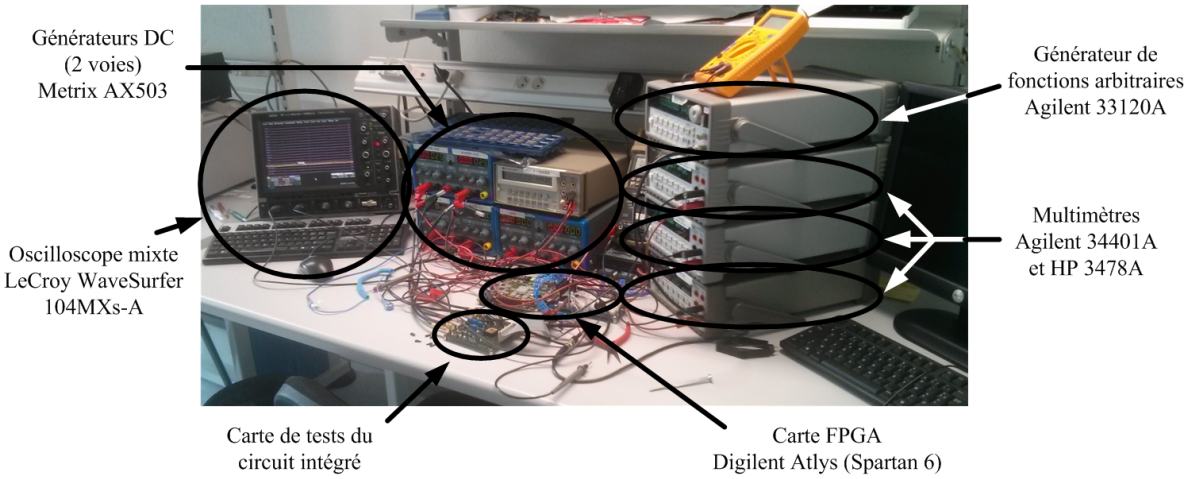

FIGURE 5.14 – Photographie du banc de test utilisé pour les mesures.

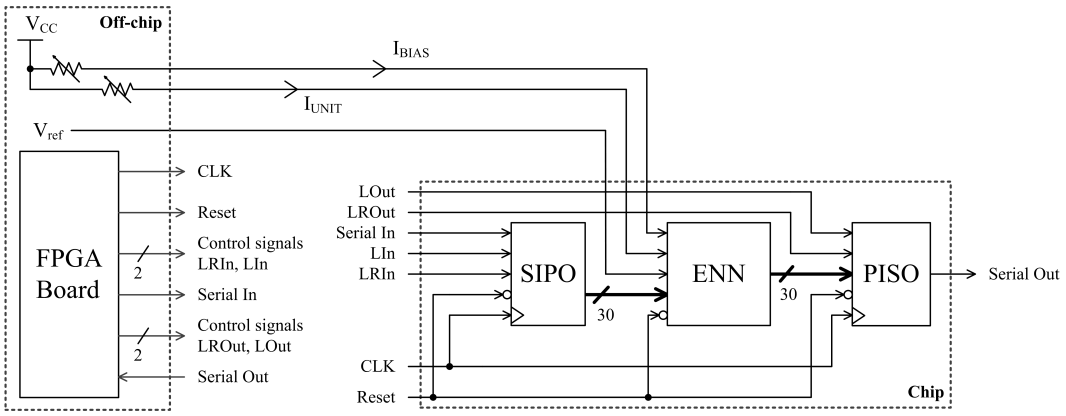

FIGURE 5.15 – Schéma de l'interfaçage du réseau avec la carte FPGA *Atlys*®.

à cette carte, ainsi que la génération de V_{ref}. Des connexions sont aussi établies avec la carte FPGA afin de transférer les signaux de commande au circuit intégré.

5.3 Mesures

5.3.1 Vérification de la fonctionnalité des éléments intégrés

5.3.1.1 Circuit "Winner-Takes-All"

Les tests de ce réseau doivent mettre en valeur :

- le bon fonctionnement du circuit WTA dans plusieurs cas de stimulations ;
- le temps de réponse d'un fanal à une stimulation ;
- le fonctionnement de la cellule de compensation des variations des paramètres environnemen-

taux ;
– si ce dernier n'est pas bon, donner la possibilité de le contourner.

La première mesure est conduite sur le réseau $R1$, afin de vérifier la fonctionnalité du circuit WTA. Les courants de polarisation sont identiques à ceux de la simulation, soit I_{UNIT} à 300 nA, I_{BIAS} à 500 nA. $Current\_Mode$ est mis à '0' et I_{MANUEL} est fixé à 280 nA, de façon à fixer de façon externe V_{ref} à 400 mV, plus ou moins 5 mV.

La mesure se déroule selon les étapes suivantes. Initialement, toutes les entrées de stimulation sont inactives. Après le début de la mesure, le fanal #1 est stimulé par l'entrée $Stim\_F1\_1\_R1$. Le fanal #1 doit alors s'activer. Le fanal #2 est ensuite stimulé par l'entrée $Stim\_F2\_R1$. Les deux fanaux sont ex-æquo, ils doivent être actifs tous les deux. Enfin, le fanal #1 reçoit une seconde stimulation par le signal $Stim\_F1\_2\_R1$. Il est alors le plus fort, le fanal #2 doit se désactiver.

Le résultat de cette mesure est montré sur la Figure 5.17. Les sorties des fanaux Out_F1_R1 et Out_F2_R1 sont visualisées à l'oscilloscope. Comme attendu, le fanal #1 s'active en premier 20 ns après stimulation. Le fanal #2 s'active ensuite après être stimulé, puis se désactive quand le fanal #1 reçoit sa seconde stimulation. Le circuit WTA se comporte donc comme prévu. L'écart entre le temps de réponse du fanal en simulation et en mesure est dû aux capacités et résistances des connexions, non prises en compte dans la simulation. De plus, lorsque la tension d'alimentation $AVCC2$ est augmentée à 1,1 V, le temps de réponse du fanal #1 est de 19,5 ns. Quand $AVCC2$ est diminuée à 0,9 V, ce temps de réponse est de 20,2 ns. Le temps de réponse est donc identique quand les conditions environnementales varient, ce qui met en avant le bon fonctionnement de la cellule de compensation.

Le Tableau 5.3 récapitule les résultats obtenus lors des mesures sur le réseau $R1$:

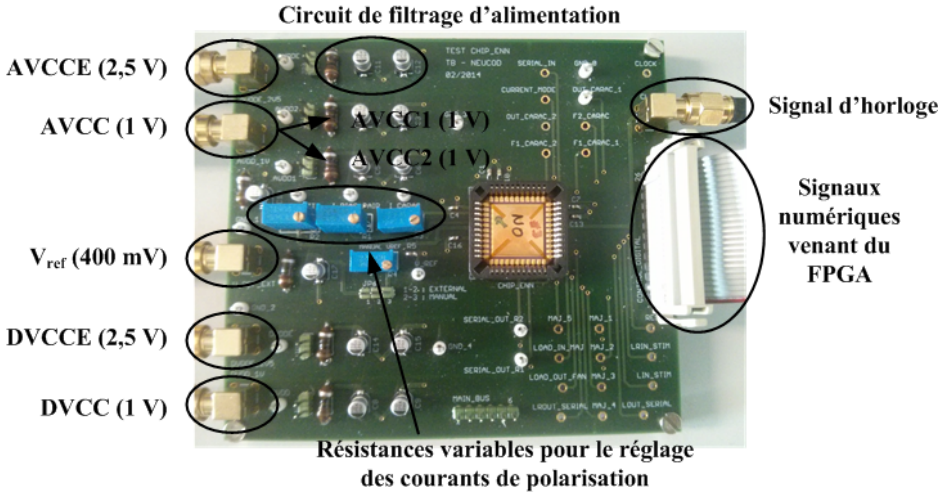

FIGURE 5.16 – Photographie de la carte de test du circuit intégré.

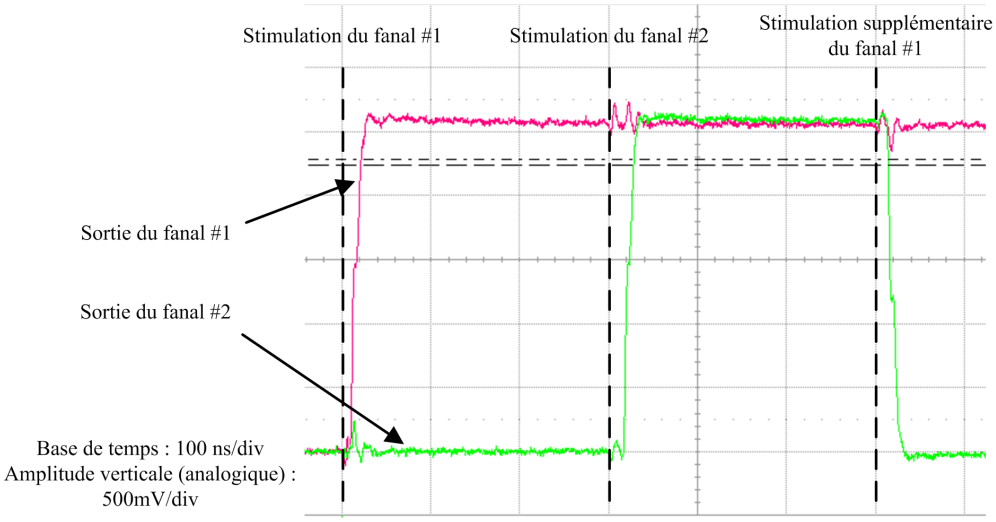

FIGURE 5.17 – Capture d'écran de l'oscilloscope montrant la réponse des fanaux #1 et #2 du réseau $R1$ à des stimulations successives.

TABLEAU 5.3 – Résultats des mesures sur $R1$ et comparaison à la simulation.

Grandeur à mesurer	Résultat obtenu en simulation	Résultat obtenu en mesure
Fonctionnalité du circuit WTA	Comparaison correcte en simulation	Fonctionnement correct, validé par la mesure
Temps de réponse d'un fanal	7,3 ns	20 ns
Fonctionnement de la cellule de compensation	Fonctionnement vérifié	Fonctionnement vérifié

5.3.1.2 Réseau complet

Une fois assuré que le circuit WTA fonctionne correctement, une mesure sur un réseau à cliques complet peut être effectuée. Cette mesure doit montrer :

- le bon fonctionnement d'un réseau à cliques complet ;
- le temps de convergence dans un cas particulier et la consommation de ce même réseau à cliques ;
- la décomposition de la récupération d'un message.

Les courants de polarisation sont fixés à 300 nA et 500 nA pour I_{UNIT} et I_{BIAS} respectivement. V_{ref} est fixé à 400 mV.

L'objectif de cette mesure est de vérifier si le réseau corrige un mot du dictionnaire légèrement modifié placé à l'entrée du réseau. Pour cela, le mot "STARA" est placé à l'entrée du réseau. La

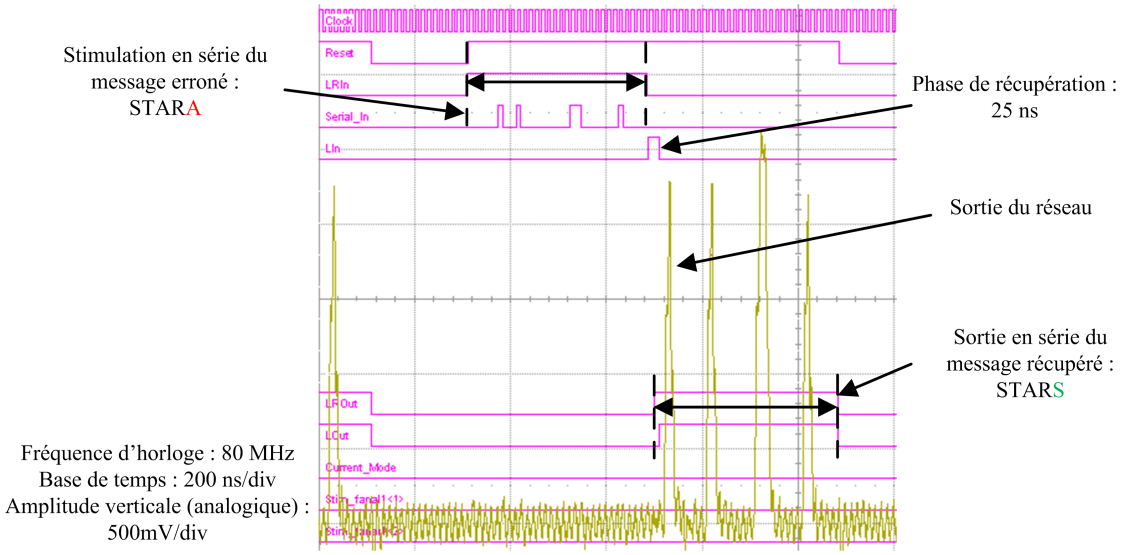

FIGURE 5.18 – Capture d'écran de l'oscilloscope montrant la réponse du réseau $R2$ à la stimulation du mot "STARA", pour une horloge de 80 MHz. Le mot est corrigé et la sortie du réseau indique "STARS".

dernière lettre est incorrecte, et le réseau devrait corriger le mot pour sortir le mot du dictionnaire "STARS". Le résultat de cette mesure est donné en Figure 5.18. Comme attendu, le réseau corrige l'erreur et on observe bien le mot "STARS" en sortie du réseau.

Le détail de chaque échange peut être visualisé en utilisant le réseau $R3$ de la même puce. Dans les mêmes conditions de mesure que pour le réseau $R2$, avec la même stimulation, on arrive aussi à corriger l'erreur, Figure 5.19. Dans ce cas, la fréquence d'horloge est diminuée à 15 MHz, afin de laisser aux buffers trois états le temps de propager les informations. On peut voir qu'au premier échange d'informations, le fanal correspondant à la lettre A du dernier cluster est actif. Lors du second échange d'informations, la forte redondance dans le réseau a corrigé l'erreur, le fanal correspondant à la lettre S du dernier cluster est actif.

Le temps de réponse peut alors être estimé en faisant varier la fréquence d'horloge, ainsi que le nombre de cycles d'horloge alloués à la phase de récupération. La Figure 5.20 montre deux mesures identiques successives sur le réseau $R2$ dans les mêmes conditions que précédemment, à l'exception de la fréquence d'horloge qui est augmentée à 100 MHz. Non seulement le réseau n'a pas convergé, puisque les deux mesures ne donnent pas les mêmes résultats, mais le temps de réaction des plots de sortie (compris entre 8,8 ns et 13,52 ns d'après la documentation du kit) est trop important pour cette fréquence de cadencement. Dans le réseau $R2$, la fréquence d'horloge maximale, c'est-à-dire avant altération de la récupération, est de 80 MHz pour une durée de convergence de deux cycles d'horloge, ou 40 MHz pour une durée de convergence d'un seul cycle d'horloge. Le temps de convergence du réseau pour cette mesure est donc de 25 ns.

La consommation est aussi mesurée lors de la phase de récupération, en mesurant séparément l'intensité du courant fourni par chaque voie d'alimentation. Sans stimulation, le cœur analogique

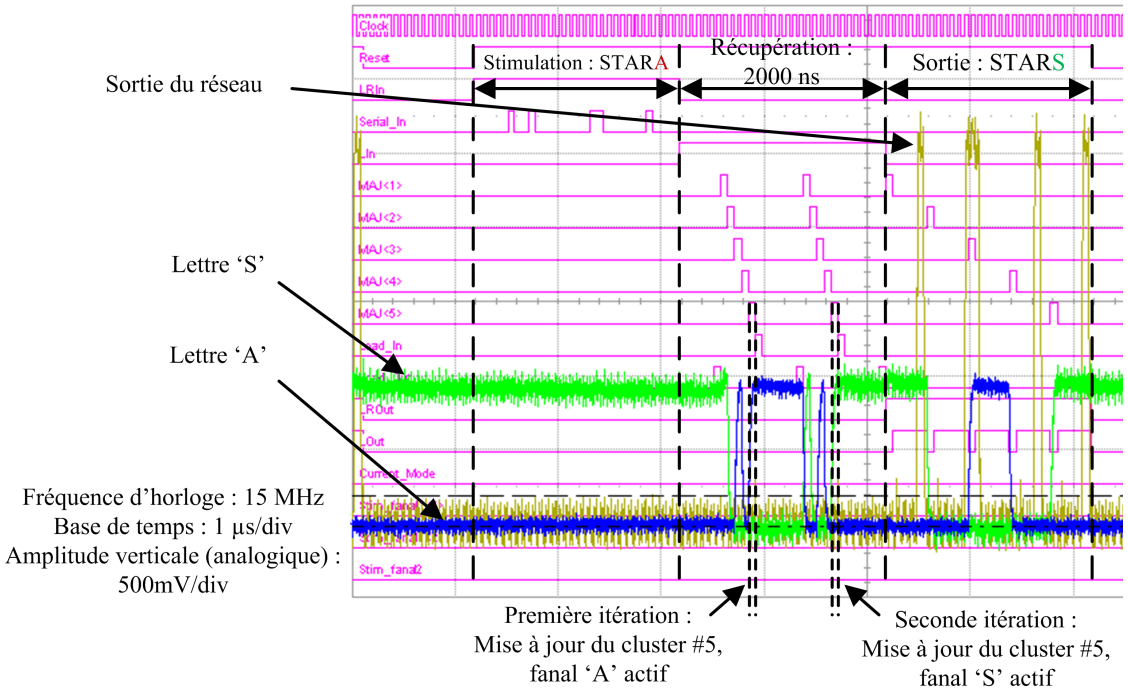

FIGURE 5.19 – Capture d’écran de l’oscilloscope montrant la réponse du réseau $R3$ à la stimulation du mot “STARA”, pour une horloge de 15 MHz. Le mot est corrigé et la sortie du réseau indique “STARS”. Le réseau $R3$ permet de visualiser les étapes intermédiaires de la récupération d’un message.

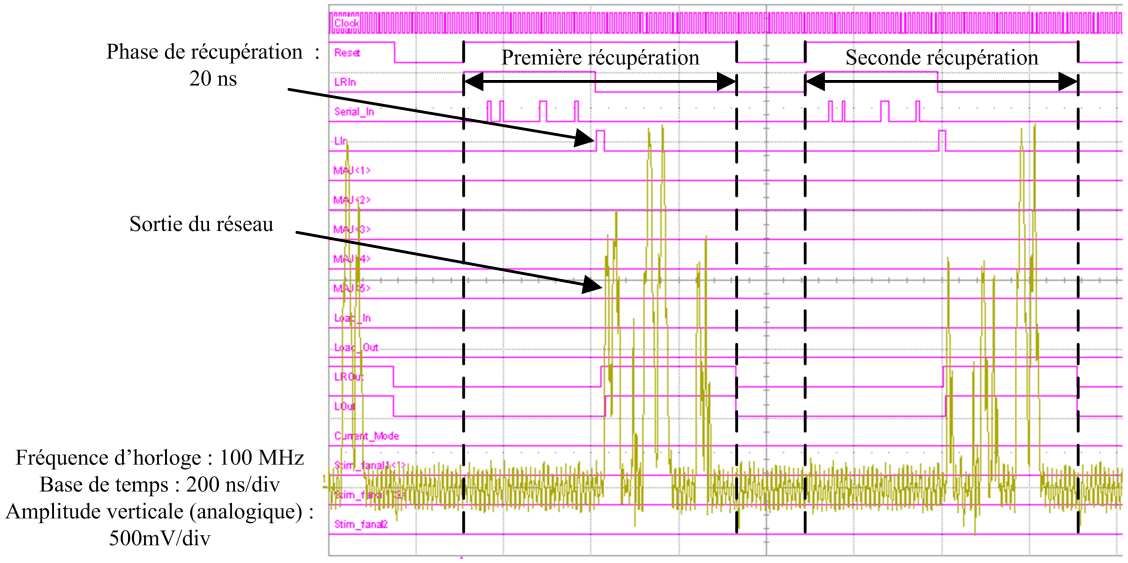

FIGURE 5.20 – Capture d’écran de l’oscilloscope montrant la réponse du réseau $R2$ à deux stimulations successives du mot “STARA”, pour une horloge de 100 MHz. Le réseau n’a pas le temps de converger dans les deux cas.

du réseau $R2$ consomme 131 μW, soit 4,3 μW par fanal. Durant la récupération d'un message, cette consommation s'élève à 145 μW pour le réseau entier. On peut considérer que dans le pire des cas, la puissance supplémentaire n'est consommée que dans les fanaux actifs, soit cinq fanaux en même temps. Un fanal actif consomme donc 7,1 μW durant la récupération d'un message, tandis qu'un fanal inactif reste à 4,3 μW. Cette consommation est légèrement supérieure à celle donnée par la simulation. Cela peut s'expliquer, par exemple, par des valeurs de courants dans les synapses supérieures à la moyenne à cause du désappariement des transistors, augmentant ainsi la consommation de puissance totale.

Le Tableau 5.4 récapitule les résultats obtenus lors de ces mesures sur les réseaux $R2$ et $R3$:

TABLEAU 5.4 – Résultats des mesures sur $R2$ et $R3$, et comparaison à la simulation.

Grandeur à mesurer	Résultat obtenu en simulation	Résultat obtenu en mesure
Fonctionnalité de l'ENN complet	Correction de "STARA" en "STARS"	Correction de "STARA" en "STARS"
Temps de convergence dans ce cas particulier	12 ns	25 ns
Consommation du réseau	3,1 μW par fanal inactif et 5,8 μW par fanal actif	4,3 μW par fanal inactif et 7,3 μW par fanal actif

5.3.2 Mesure des performances de récupération d'information

Le fonctionnement du réseau complet étant vérifié, ses performances en terme de pouvoir de correction doivent être évaluées :

- l'allure de son pouvoir de récupération d'information ;
- le temps de convergence maximal du réseau de neurones à cliques.

Pour cela, on effectue le même test que pour faire l'évaluation théorique, cf. Chapitre 4. Les 16 807 différentes stimulations sont générées les unes après les autres par le FPGA. Elles sont successivement entrées dans le réseau, qui effectue une récupération de message après l'autre. Quand le réseau donne son résultat, il le fournit aussi au FPGA qui compare le mot décodé aux mots dans le dictionnaire. Il classe alors ce résultat en fonction du nombre d'erreurs insérées dans le mot stocké. Ces tests sont effectués sur 22 puces, trois ayant été endommagées lors des mesures.

La Figure 5.21 donne la courbe correspondant au nombre de récupérations erronées fournies par le réseau, en fonction du nombre d'erreurs introduites dans la stimulation. Cette courbe représente la moyenne des tests effectués sur les 22 puces fonctionnelles. L'écart-type autour de chaque point varie entre 1,8% et 3%. Cette courbe est comparée à la courbe de taux d'erreur théorique obtenue en simulation par un modèle du circuit, cf. Chapitre 4. Les résultats de mesures sont éloignés des résultats de simulation idéaux, avec un écart maximum de 12% obtenus lorsque deux erreurs sont

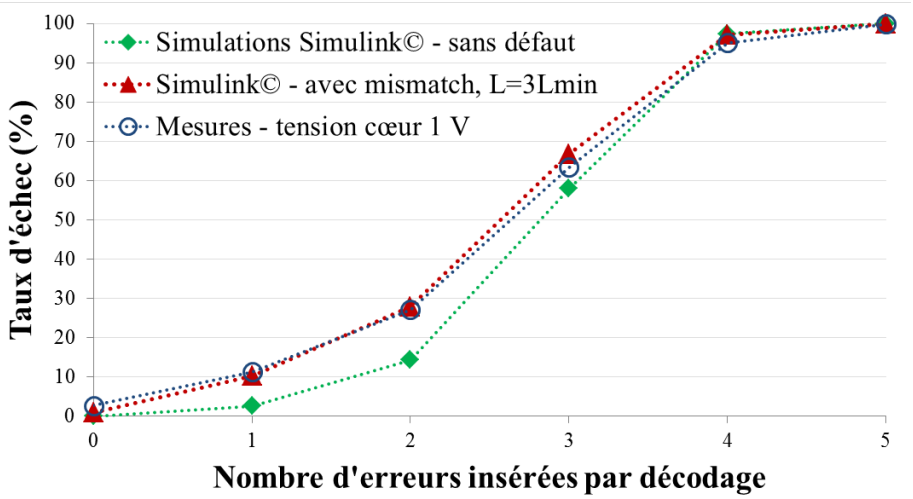

FIGURE 5.21 – Taux d'erreur du réseau en fonction du nombre d'erreurs introduites dans la stimulation. Les tests sont réalisés en simulation *Simulink*®, avec et sans défaut d'appariement des transistors, et par la mesure.

insérées dans le message original.

En revanche, en superposant la courbe de mesures à celle simulant le circuit incluant un défaut d'appariement des transistors, on s'aperçoit que ce problème explique bien l'écart de la mesure à la simulation idéale. Les techniques de layout mises en place dans la conception des masques du circuit n'ont pas réduit l'impact du défaut d'appariement des transistors. Notamment, les grilles de transistors factices mises en place doivent être remplacées par des transistors factices. Ceux-ci permettent de mieux limiter les défauts d'appariement des transistors, d'après [CRD+02].

Le temps de réponse maximal de la récupération d'un message est aussi estimé grâce à ce test. Pour cela, la phase de récupération, qui a une durée T_{CONV}, est réduite à une durée d'une période d'horloge. En faisant varier la fréquence d'horloge, on maîtrise alors la valeur de T_{CONV} et on peut estimer à partir de quelle valeur les performances sont réduites. Les résultats de ce test est montré sur la Figure 5.22.

Pour des valeurs de T_{CONV} supérieures à 58 ns, les performances du réseau ne changent pas et restent proches de celles de la simulation idéale. En revanche, en-dessous de 58 ns, les performances du réseau diminuent car ce dernier n'a pas le temps de converger pour tous les cas de stimulation. Le temps de convergence maximal du réseau T_{CONV} est donc de 58 ns.

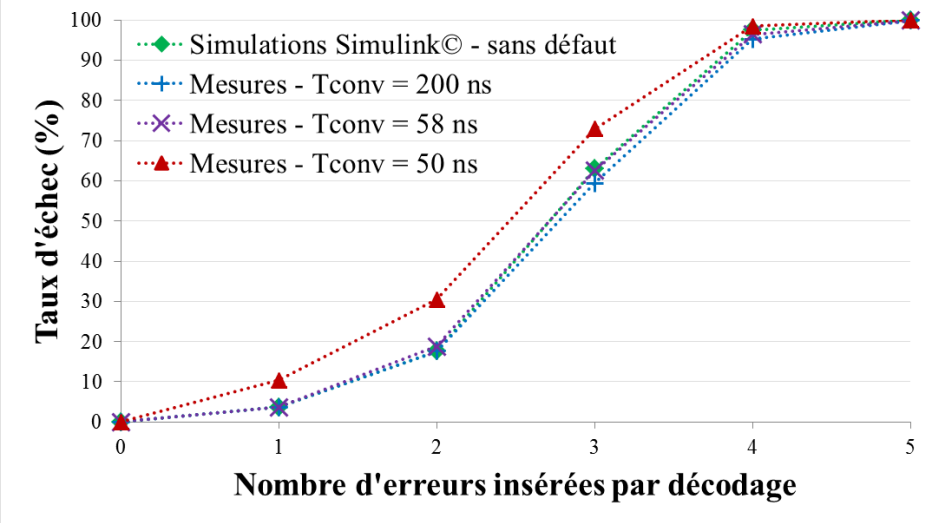

FIGURE 5.22 – Taux d'erreur du réseau en fonction du nombre d'erreurs introduites dans la stimulation. Les tests sont réalisés par la mesure pour plusieurs valeurs de T_{CONV}, et comparées aux performances de la simulation idéale.

Le Tableau 5.5 récapitule les résultats obtenus lors de ces mesures de performances :

TABLEAU 5.5 – Résultats des mesures de performances du réseau de neurones à cliques, et comparaison à la simulation.

Grandeur à mesurer	Résultat obtenu en simulation	Résultat obtenu en mesure
Pouvoir de récupération d'information	voir Figure 5.21, courbes verte et rouge	voir Figure 5.21, courbe bleue
Temps de convergence maximal	21 ns	58 ns

5.3.3 Variation des conditions de mesure

L'objectif de cette section est de vérifier que les variations des paramètres environnementaux sont compensées par le dispositif montré dans le Chapitre 4.

Pour cela, les mesures sont effectuées avec des tensions d'alimentation variant de 10% de leur valeur nominale, qui est 1 V. Les courants de polarisation du circuit étant générés à partir de l'alimentation analogique, leurs valeurs sont modifiées par cette variation. Leurs valeurs sont alors réajustées grâce aux résistances variables sur la carte de test.

Les mesures sont conduites pour des tensions d'alimentation analogique et numérique de 0,9 V et 1,1 V. Les résultats de ces tests, en termes de performances de récupération d'information, sont

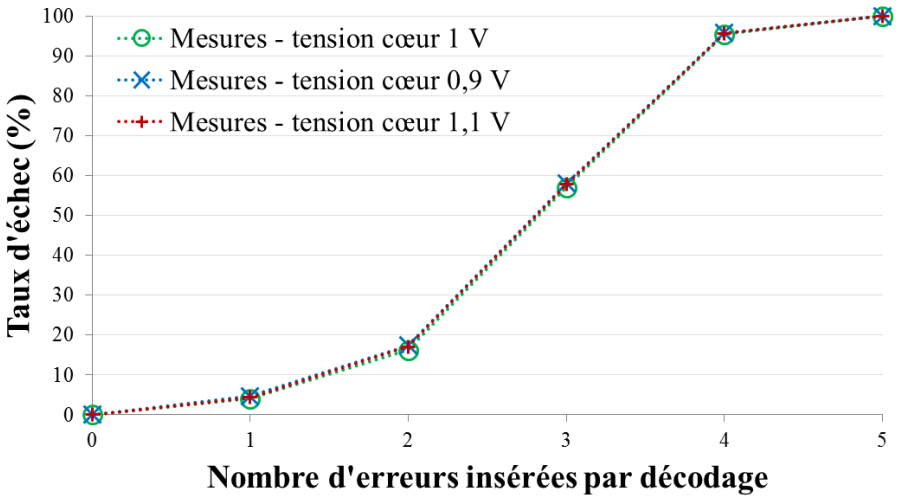

FIGURE 5.23 – Taux d'erreur du réseau en fonction du nombre d'erreurs introduites dans la stimulation. Les tests sont réalisés par la mesure pour plusieurs valeurs de la tension d'alimentation du cœur de la puce.

FIGURE 5.24 – Photographie de l'étuve utilisée pour les mesures avec variations de température.

donnés par la Figure 5.23. Les courbes de performances sont très proches, l'écart maximal entre ces courbes étant de 1,3%. Les variations de la tension d'alimentation n'ont donc pas d'influence sur le circuit, grâce au dispositif de compensation.

L'impact des variations de température sur le circuit est aussi étudié. Une étuve permettant régler la température dans une enceinte close est utilisée afin de contrôler la température du circuit, Figure 5.24. Pour des raisons pratiques et en raison de la disponibilité du matériel, des mesures sur la seule puce #13 ont été effectuées. De même, des variations combinées de la température et des

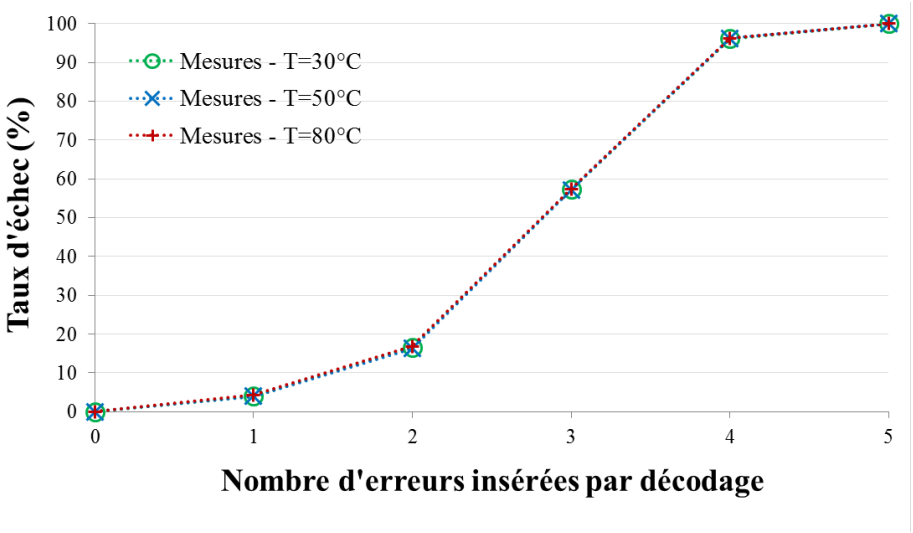

FIGURE 5.25 – Taux d'erreur du réseau en fonction du nombre d'erreurs introduites dans la stimulation. Les tests sont réalisés par la mesure sur la puce #13, pour plusieurs valeurs de la température.

tensions d'alimentation du circuit n'ont pas pu être mises en place.

Les tests sont donc effectués sur la puce #13 en faisant varier la température de 25 ˚C à 80 ˚C, par pas de 5 ˚C. Les résultats sont montrés sur la Figure 5.25. Les courbes montrent l'évolution du taux d'erreur de la récupération d'information en fonction de la température du circuit, pour chaque cas de stimulation. Comme le montre la Figure 5.25, les performances de récupération d'information sont très peu affectées par la température du circuit. L'écart maximal avec les conditions typiques de fonctionnement, c'est-à-dire une température de 25 ˚C, est de 0,3%.

Le circuit conçu avec un dispositif de compensation des paramètres environnementaux est donc bien robuste à une variation de ces derniers. Les performances obtenues par le circuit présenté dans ce document sont donc rendues indépendantes des conditions environnementales dans lesquelles se déroulent les tests.

5.4 Comparaison à une implantation numérique des réseaux de neurones à cliques

Nous cherchons finalement à comparer les performances obtenues lors des mesures par le circuit que nous avons développé à celles d'un circuit numérique équivalent. Afin que cette comparaison soit équitable, nous visons une implantation numérique sur ASIC, en utilisant le même kit de conception que notre circuit, à savoir le kit ST CMOS 65 nm. Le circuit numérique utilise donc les cellules numériques de référence fournies par la bibliothèque de composants du kit de conception.

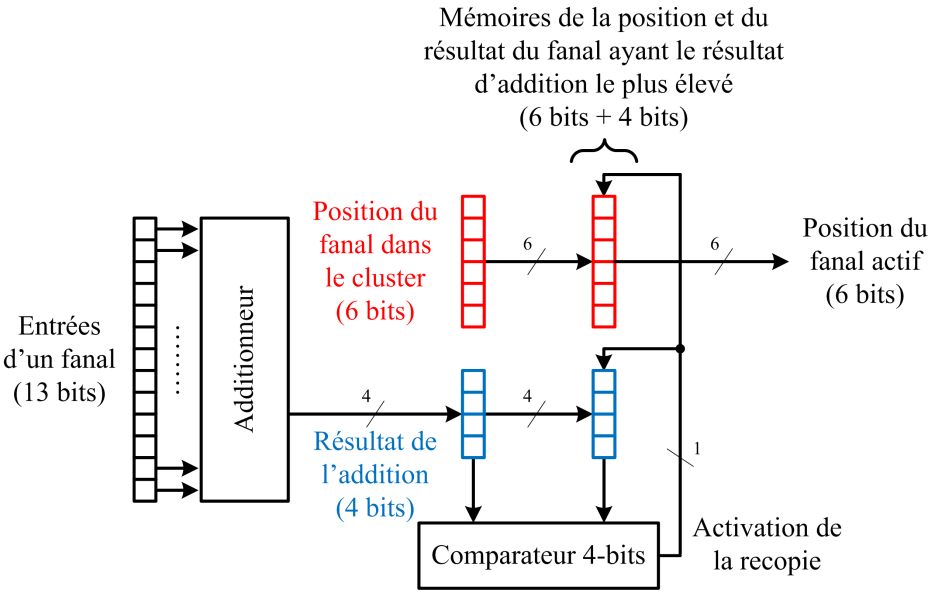

FIGURE 5.26 – Schéma du circuit numérique d'un cluster.

5.4.1 Description du circuit numérique

L'architecture du circuit numérique s'appuie sur l'architecture d'un cluster conçue dans [LLAS13] à des fins de comparaison d'implantations analogique et numérique des réseaux de neurones à cliques. Celle-ci est rappelée dans la Figure 5.26 et adaptée au cas d'étude. Les entrées d'un fanal sont codées sur treize bits, qui sont l'équivalent des treize synapses connectées à chaque fanal dans l'implantation analogique. Les bits d'entrée sont ensuite fournis à un additionneur, constitué d'une chaîne d'additionneurs 1-bit, qui donne leur somme sur quatre bits en un cycle d'horloge. On associe à ces quatre bits six autres bits représentant la position du fanal dans le cluster. Un comparateur 4-bits compare ensuite en un cycle d'horloge chaque résultat de l'additionneur avec le résultat le plus élevé en mémoire. Si le dernier résultat de l'additionneur est supérieur, un bit d'activation de recopie est mis à '1' pour permettre la recopie de ce résultat dans une mémoire, en tant que nouveau résultat le plus élevé. En itérant ce processus pour tous les fanaux d'un cluster, on détermine le fanal actif dans ce cluster et on fournit sa position à tous les autres clusters.

Cette architecture se base sur la réutilisation matérielle au sein d'un cluster. Pendant un cycle d'horloge, une comparaison est effectuée entre les résultats d'addition de deux fanaux, les entrées du fanal suivant sont additionnées et les bits d'entrée du prochain fanal du cluster sont placés en entrée de l'additionneur. Ainsi, le fanal actif dans un cluster est déterminé en huit périodes d'horloge. De plus, la communication entre les clusters est assurée en utilisant la méthode de communication numérique de l'organisation numérique multi-clusters décrite dans la Section 2.2.3.2.1. En utilisant l'équation donnée dans le Tableau 2.7 pour cette organisation, la latence entre deux itérations est de 180 cycles d'horloge dans notre cas d'étude.

Enfin, les dix mots du dictionnaire du Tableau 5.2 sont stockés dans la mémoire des connexions

du circuit numérique. Afin, de ne pas biaiser la comparaison avec le circuit intégré proposé, la mémoire des connexions ne contient que les informations sur les connexions réalisées effectivement dans le réseau. Le réseau n'est alors plus flexible.

5.4.2 Performances du circuit numérique

La structure du circuit numérique est décrite en langage VHDL, puis la synthèse ASIC est réalisée en utilisant le logiciel *Synopsis Design Compiler*®. En incluant les registres SIPO et PISO permettant de faire l'interface en série avec l'extérieur du circuit, identiques à ceux présentés dans la Section 5.2.1.2, le circuit occupe une surface de silicium de 10 621 μm^2.

Pour déterminer le temps de convergence du réseau, on considère un nombre d'itérations du processus de récupération des messages fixe. Dans la plupart des cas, quatre itérations suffisent pour corriger les erreurs. Le réseau a donc besoin de 572 cycles d'horloge en tout pour converger. La fréquence de fonctionnement du circuit étant évaluée à 518 MHz, le temps de convergence T_{CONV} du circuit est donc de 1 104 ns.

Les performances du circuit numérique estimées après synthèse par *Synopsis Design Compiler*® pour la technologie ST 65 nm sont rappelées dans le Tableau 5.6, et comparées aux performances du réseau $R2$ sur le circuit intégré.

TABLEAU 5.6 – Comparaison des caractéristiques des implantations numériques et analogiques du réseau de neurones à cliques.

Paramètre	Implantation numérique	Réseau $R2$	Ratio numérique/analogique
Surface de silicium occupée (μm^2)	10 621	19 338	$\div$1.82
Temps de convergence maximal (ns)	1 104	58	$\times$19.03

L'implantation numérique occupe près de deux fois moins de surface de silicium pour l'intégration des mêmes fonctions que dans le circuit intégré. Bien que les fonctions mises en œuvre dans le circuit numérique soient plus complexes que leur implantation analogique, l'utilisation de cellules élémentaires (faites de transistors de tailles minimales) ainsi que la réutilisation matérielle au sein des clusters permettent d'obtenir un circuit plus compact. En revanche, cette compacité se paie par un temps de convergence 19 fois plus grand que dans l'implantation analogique. En considérant que les données traitées sont identiques pour les deux implantations, le débit de données traitées est donc 19 fois supérieur pour le circuit analogique.

Nous définissons l'efficacité d'un circuit comme le rapport entre la quantité de données traitées et le coût pour les traiter en termes de surface de silicium et de temps ici. L'efficacité telle que nous

la définissons est donc le rapport entre le débit de traitement d'un circuit et sa surface de silicium occupée. Nous ne nous intéressons pas à la consommation d'énergie ici, car nous ne l'avons pas estimée précisément dans le cas d'une implantation numérique sur ASIC. Comme les deux circuits sont conçus pour traiter les mêmes données, l'efficacité du circuit analogique est dix fois supérieure à celle d'un équivalent numérique.

Il est donc possible de traiter des données pour un plus faible coût de surface de silicium et latence combinées avec une implantation analogique des réseaux de neurones à cliques qu'avec une implantation entièrement numérique.

Conclusion

Ce chapitre a présenté le premier ASIC intégrant un circuit de réseaux de neurones à cliques, ainsi que le banc de tests permettant de réaliser les mesures. En fonction des mesures à réaliser pour caractériser le circuit, trois réseaux sont intégrés et permettent de donner les paramètres suivant du circuit : temps de réponse d'un fanal, temps de convergence du réseau, fonctionnalité du circuit WTA et du circuit de compensation des variations des paramètres environnementaux, pouvoir de récupération d'information et consommation du circuit.

Les résultats obtenus lors des mesures corroborent ceux obtenus en simulation dans le Chapitre 4, au niveau du pouvoir de récupération d'information. Les différences observées en terme de temps de réponse d'un fanal et de temps de convergence du réseau sont expliquées par les éléments passifs dus aux connexions, non représentés en simulation. Une perspective future est donc naturellement l'optimisation de la conception des masques, afin de réduire la latence du réseau et se rapprocher des résultats de simulation à ce niveau.

Le circuit produit est de plus robuste dans différentes conditions environnementales, sans perte de pouvoir de récupération d'information. Ce résultat valide l'intérêt de l'utilisation de la cellule de compensation des variations des paramètres environnementaux.

Enfin, le circuit produit est comparé à un équivalent numérique réalisé à l'aide d'une synthèse ASIC. Les performances de ce dernier montrent que l'implantation analogique des réseaux de neurones à cliques est dix fois plus efficace qu'une implantation numérique, en termes de surface de silicium et de latence.

Conclusion et perspectives

Dans ce document, nous avons montré qu'il était possible de réaliser un circuit capable d'intégrer un grand nombre de neurones, ou fanaux dans notre cas. Nous avons également minimisé la surface de silicium occupée par le réseau grâce à des architectures mixtes analogiques/numériques et des techniques de conception de circuits comme le calcul en mode courant. Nous avons enfin prouvé le fonctionnement et la robustesse des concepts que nous avons avancés sur un réseau de taille réduite que nous avons intégré sur puce puis testé.

Dans le **Chapitre 1**, après avoir rappelé quelques notions sur les réseaux de neurones, nous avons fait l'inventaire des modèles de réseaux de neurones existants et des techniques d'intégration de ces réseaux sur circuit. Compte tenu de nos objectifs, qui étaient de concevoir un circuit capable d'intégrer un réseau de centaines voire de milliers de neurones, nous avons décidé d'utiliser le modèle des réseaux de neurones à cliques décrit dans [GB11]. Ces réseaux, comparés aux autres modèles, utilisent des fonctions simples dans le cœur des neurones, appelés fanaux, et ont une connectivité limitée entre ces derniers. Nous avons de plus fait le choix d'intégrer ce modèle de réseau de manière bio-inspirée, c'est-à-dire en faisant abstraction du comportement physiologique d'un neurone dans les échanges d'information.

Nous avons ensuite défini les choix technologiques adoptés dans ce document, notamment l'utilisation du kit de conception ST CMOS 65 nm, dans le **Chapitre 2**. Nous avons proposé une architecture mixte analogique/numérique générique qui permet de construire simplement un réseau de n'importe quelle taille grâce à l'utilisation d'un motif de base facilement modulable et connectable dans le cas d'un réseau entièrement parallèle. Nous avons alors comparé les augmentations de surface de cette architecture adaptée au réseau à cliques, puis à un réseau complètement connecté, comme un réseau de Hopfield. Un réseau à cliques a une dépendance quadratique de la surface par rapport au nombre de neurones, contre une dépendance cubique pour un réseau de Hopfield. Une fois l'architecture définie, nous avons cherché à minimiser la surface qu'elle occupe en utilisant la réutilisation matérielle que permet un système de communication numérique. Nous sommes donc arrivés à un compromis entre la surface de silicium occupée et la latence du circuit.

Les éléments intégrant les fonctions de base d'un fanal ont été décrits dans le **Chapitre 3**. En utilisant le mode courant pour intégrer les fonctions de calcul, nous avons simplifié le circuit d'un fanal, notamment l'addition des contributions qui est réduite à un simple nœud. Nous avons ensuite présenté différents circuits intégrant les règles de comparaison définies dans le modèle, les

règles WTA, WsTA et LsKO. Finalement, nous avons donné le circuit d'un fanal complet, puis d'un cluster utilisant la règle de comparaison WTA locale au cluster.

Dans le **Chapitre 4**, nous avons présenté les résultats de simulation des concepts sur un réseau de trente fanaux. Tout d'abord, nous avons défini les paramètres selon lesquels nous avons évalué les performances du réseau que nous cherchons à intégrer : pouvoir de récupération d'information, temps de convergence et consommation du réseau. Nous avons ensuite simulé le réseau grâce à un modèle haut niveau pour avoir les valeurs des paramètres de performance du réseau. Nous avons alors vérifié la fonctionnalité des principes mis en œuvre dans le modèle en simulation. Dans la suite du chapitre, nous avons cherché à vérifier la robustesse du circuit en vue de l'intégration sur puce. Nous avons pour cela considéré deux aspects, les variations des conditions de l'environnement dans lequel fonctionnent les transistors et le désappariement des transistors. Après avoir simulé les effets de ces imperfections sur le fonctionnement global du réseau, nous avons mis en place des dispositifs de compensation de ces imperfections pour rester le plus proche possible des performances du circuit simulées dans des conditions idéales. Le critère de robustesse est donc vérifié.

Finalement, le **Chapitre 5** nous a permis de montrer les résultats des tests du circuit intégré fabriqué. Nous avons tout d'abord présenté les objectifs que nous avons cherché à atteindre avec les résultats de ces tests, notamment retrouver les performances du réseau simulé dans le Chapitre 4. En fonction de ces objectifs, nous avons décrit les circuits que nous avons intégrés sur puce, ainsi que les tests permettant de mettre en valeur les résultats attendus. Le fonctionnement du réseau à cliques est vérifié lors des mesures effectuées. Le réseau de trente fanaux, intégré dans un circuit de 16 470 μm^2, a un pouvoir de décodage correspondant à celui simulé pour un circuit incluant le désappariement des transistors. Le réseau a aussi un temps de réponse de 58 ns, soit plus du double de celui obtenu en simulation, à cause des résistances parasites dans les connexions ignorées en simulation. La consommation d'un fanal est de 4,3 μW pour un fanal inactif et de 7,1 μW pour un fanal actif. La consommation visée de l'ordre de grandeur du microwatt par fanal définie dans l'introduction de ce document est donc atteinte. Nous avons démontré la robustesse du circuit, dont les performances ne changent pas dans des conditions environnementales d'opération des transistors différentes. Enfin, nous avons montré que le circuit produit est dix fois plus efficace qu'un équivalent numérique en termes de surface de silicium occupée et de latence.

Afin de poursuivre le travail entrepris dans ce document, des pistes de développement peuvent être explorées. Tout d'abord, il est nécessaire d'apporter de la flexibilité pour la gestion du dictionnaire sur le circuit intégré. Des éléments ont été proposés pour permettre des modifications du dictionnaire du réseau, comme l'ajout d'un bit d'activation de chaque synapse, proposé dans le Chapitre 3, Section 3.2.3.3. Ces bits d'activation peuvent ainsi être stockés dans une mémoire gérant les connexions dans le réseau, comme décrit dans le Chapitre 2, Section 2.2.3.2. Ces deux solutions n'ont toutefois pas été intégrées sur puce, et leur fonctionnement sur circuit doit donc être testé.

Ensuite, la prochaine étape est d'intégrer effectivement un réseau de grande taille, c'est-à-dire des centaines de fanaux, sur puce. Pour limiter l'augmentation de la surface de silicium occupée par

le réseau lorsque le nombre de fanaux augmente, nous avons défini dans le Chapitre 2, Section 2.2.3.2 deux architectures de réseau se basant sur la réutilisation matérielle. L'intégration de l'une de ces solutions sur puce est donc une étape vers la réalisation d'un circuit intégré contenant un réseau de plusieurs centaines de fanaux. De plus, des travaux ont montré dans [LLSA15] qu'il était possible de diminuer encore plus la surface d'un fanal, en utilisant des transistors de taille minimale fonctionnant en régime sous le seuil. La surface d'un fanal passe alors de 17,7 μm^2 à 9,5 μm^2, et celle d'une synapse de 7,5 μm^2 à 3,6 μm^2. Cependant, nous avons vu dans cette étude que les transistors de taille minimale sont plus vulnérables au désappariement que des transistors de plus grandes dimensions. De plus, en diminuant la tension d'alimentation et les courants unitaires, nous arrivons à diminuer la puissance consommée par un fanal, mais en augmentant son temps de réponse. L'influence sur la consommation d'énergie du réseau n'est donc pas évidente. Nous avons déterminé que pour une tension d'alimentation de 0,6 V et un courant unitaire de 50 nA, l'énergie consommée par un fanal peut être réduite jusqu'à 32 fJ, en simulation. Les performances d'un tel circuit sont donc un compromis entre le taux de récupération de messages à atteindre, la quantité d'information stockée dans le réseau, la surface du circuit et sa consommation d'énergie. Nous avons commencé à traiter ces aspects, et il est intéressant de continuer ces travaux, qui sont un pas de plus vers la réalisation d'un réseau contenant un grand nombre de fanaux sur puce et consommant peu d'énergie.

Une fois qu'un réseau flexible de grande taille pourra être intégré sur puce, ce dernier pourra être utilisé dans des domaines d'application variés. Des travaux ont déjà été entamés pour utiliser des circuits implantant des réseaux de neurones à cliques en tant que mémoires associatives dans [JGOG15]. De plus, le circuit proposé dans ce document a été utilisé dans un système de gestion de la puissance dans les MPSoCs (Multiprocessor System-on-Chip) dans [LBL$^+$13] et [LBL$^+$14]. Des travaux sont également en cours pour réaliser des traitements de signaux bio-médicaux sur puce grâce à un réseau de neurones à cliques.

Enfin, une autre possibilité d'évolution du circuit proposé est d'intégrer les autres règles d'activation d'un fanal définies dans le Chapitre 1, Section 1.3.3.2. Dans ce document, nous ne considérons en effet que l'intégration sur circuit de la règle d'activation WTA. Cependant, il est nécessaire d'utiliser des règles globales d'activation dans certains cas, notamment pour des messages parcimonieux (dans lesquels les cliques ne relient pas un fanal de chaque cluster du réseau, mais d'un sous-ensemble de clusters). Par exemple, une application comme la classification de signaux bio-médicaux utilise un réseau de neurones contenant des messages parcimonieux. Toutefois, la complexité des implantations de ces règles d'activation des fanaux doit être étudiée car ces dernières s'appliquent non plus à un groupe de fanaux, mais à l'ensemble des fanaux présents dans le réseau. Cela ouvre des perspectives vers la conception d'un circuit polyvalent capable d'implanter plusieurs règles de décodage de manière efficace en termes de surface de silicium et de consommation d'énergie.

Bibliographie

[Abb99] Larry F Abbott. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain research bulletin*, 50(5) :303–304, 1999.

[ABGJ14] B.K. Aliabadi, C. Berrou, V. Gripon, and Xiaoran Jiang. Storing sparse messages in networks of neural cliques. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(5) :980–989, May 2014.

[ABP+91] A.G. Andreou, K.A. Boahen, P.O. Pouliquen, A. Pavasovic, R.E. Jenkins, and K. Strohbehn. Current-mode subthreshold MOS circuits for analog VLSI neural systems. *Neural Networks, IEEE Transactions on*, 2(2) :205–213, Mar 1991.

[AF96] A.P. Almeida and J.E. Franca. Digitally programmable analog building blocks for the implementation of artificial neural networks. *Neural Networks, IEEE Transactions on*, 7(2) :506–514, Mar 1996.

[AGJ14] Ala Aboudib, Vincent Gripon, and Xiaoran Jiang. A study of retrieval algorithms of sparse messages in networks of neural cliques. In *COGNITIVE 2014 : The Sixth International Conference on Advanced Cognitive Technologies and Applications*, pages 140–146, 2014.

[BGSH14] B. Boguslawski, V. Gripon, F. Seguin, and F. Heitzmann. Huffman coding for storing non-uniformly distributed messages in networks of neural cliques. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 262–268, July 2014.

[BiPM04] A. Bofill-i Petit and A.F. Murray. Synchrony detection and amplification by silicon neurons with STDP synapses. *Neural Networks, IEEE Transactions on*, 15(5) :1296–1304, Sept 2004.

[CCWC15] Philippe Coussy, Cyrille Chavet, Hugues Nono Wouafo, and Laura Conde-Canencia. Fully binary neural network model and optimized hardware architectures for associative memories. *JETC*, 11(4) :35, 2015.

[CFSV11] G. Carvajal, M. Figueroa, D. Sbarbaro, and W. Valenzuela. Analysis and compensation of the effects of analog VLSI arithmetic on the LMS algorithm. *Neural Networks, IEEE Transactions on*, 22(7) :1046–1060, July 2011.

[CGD+07] P. Camilleri, M. Giulioni, V. Dante, D. Badoni, G. Indiveri, B. Michaelis, J. Braun, and P. Del Giudice. A neuromorphic aVLSI network chip with configurable plastic synapses.

In *Hybrid Intelligent Systems, Proceedings of the 7th International Conference on*, pages 296–301, 2007.

[CRD+02] J.A. Croon, M. Rosmeulen, S. Decoutere, Willy Sansen, and H.E. Maes. An easy-to-use mismatch model for the MOS transistor. *Solid-State Circuits, IEEE Journal of*, 37(8) :1056–1064, Aug 2002.

[CSBI14] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri. Neuromorphic electronic circuits for building autonomous cognitive systems. *Proceedings of the IEEE*, 102(9) :1367–1388, Sept 2014.

[CZR+14] F. Corradi, D. Zambrano, M. Raglianti, G. Passetti, C. Laschi, and G. Indiveri. Towards a neuromorphic vestibular system. *Biomedical Circuits and Systems, IEEE Transactions on*, 8(5) :669–680, Oct 2014.

[DST98] A. Demosthenous, S. Smedley, and J. Taylor. A CMOS analog winner-take-all network for large-scale applications. *Circuits and Systems I : Fundamental Theory and Applications, IEEE Transactions on*, 45(3) :300–304, Mar 1998.

[Fit61] Richard FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6) :445, 1961.

[GB11] V. Gripon and C. Berrou. Sparse neural networks with large learning diversity. *Neural Networks, IEEE Transactions on*, 22(7), July 2011.

[GC12] Ming Gu and S. Chakrabartty. Synthesis of bias-scalable CMOS analog computational circuits using margin propagation. *Circuits and Systems I : Regular Papers, IEEE Transactions on*, 59(2) :243–254, Feb 2012.

[GTBB09] L. Gatet, H. Tap-Beteille, and F. Bony. Comparison between analog and digital neural network implementations for range-finding applications. *Neural Networks, IEEE Transactions on*, 20(3) :460–470, March 2009.

[GYMD12] I.C. Goknar, M. Yildiz, S. Minaei, and E. Deniz. Neural CMOS-integrated circuit and its application to data classification. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(5) :717–724, May 2012.

[Heb49] Donald Olding Hebb. *The organization of behavior : A neuropsychological approach.* John Wiley & Sons, 1949.

[HH52] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4) :500–544, 1952.

[Hop82] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of National Academy of Sciences, Biophysics*, pages 2554–2558, April 1982.

[HT12] Hung-Yi Hsieh and Kea-Tiong Tang. VLSI implementation of a bio-inspired olfactory spiking neural network. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(7) :1065–1073, July 2012.

[ICD06] G. Indiveri, E. Chicca, and R. Douglas. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *Neural Networks, IEEE Transactions on*, 17(1), January 2006.

[JBTH12] A. Joubert, B. Belladj, O. Temam, and R. Héliot. Hardware spiking neurons design : analog or digital ? In *Computational Intelligence (WCCI), IEE World Congress on*, June 2012.

[JGOG15] H. Jarollahi, V. Gripon, N. Onizawa, and W.J. Gross. Algorithm and architecture for a low-power content-addressable memory based on sparse clustered networks. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 23(4) :642–653, April 2015.

[JOGG12] H. Jarollahi, N. Onizawa, V. Gripon, and W.J. Gross. Architecture and implementation of an associative memory using sparse clustered networks. In *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, pages 2901–2904, May 2012.

[JOGG14] Hooman Jarollahi, Naoya Onizawa, Vincent Gripon, and Warren J Gross. Algorithm and architecture of fully-parallel associative memories based on sparse clustered networks. *Journal of Signal Processing Systems*, 76(3) :235–247, 2014.

[KABH06] R. J. Kier, J. C. Ames, R. D. Beer, and R. R. Harrison. Design and implementation of multipattern generators in analog VLSI. *Neural Networks, IEEE Transactions on*, 17(4), July 2006.

[LBBH98] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, Nov 1998.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553) :436–444, May 2015.

[LBL+13] B. Larras, B. Boguslawski, C. Lahuec, M. Arzel, F. Seguin, and F. Heitzmann. Analog encoded neural network for power management in MPSoC. In *New Circuits and Systems Conference (NEWCAS), 2013 IEEE 11th International*, pages 1–4, June 2013.

[LBL+14] Benoit Larras, Bartosz Boguslawski, Cyril Lahuec, Matthieu Arzel, Fabrice Seguin, and Frédéric Heitzmann. Analog encoded neural network for power management in MPSoC. *Analog Integrated Circuits and Signal Processing*, 81(3) :595–605, 2014.

[LLAS13] B. Larras, C. Lahuec, M. Arzel, and F. Seguin. Analog implementation of encoded neural networks. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 1612–1615, May 2013.

[LLSA15] B. Larras, C. Lahuec, F. Seguin, and M. Arzel. Design of analog subthreshold encoded neural networks circuit in sub-100nm CMOS. In *Neural Networks, 2015 (IJCNN 2015), International Joint Conference on*, 2015.

[LOZ06] L. Lin, R. Osan, and Tsien J. Z. Organizing principles of real-time memory encoding : Neural clique assemblies and universal neural codes. *Trends in Neuroscience*, 29(1) :48–57, Jan 2006.

[LRMM88] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, and C. A. Mead. Winner-take-all networks of o(n) complexity. Technical report, CALIFORNIA INST OF TECH PASADENA DEPT OF COMPUTER SCIENCE, 1988.

[MB99] W. Maass and C. Bishop. *Analog VLSI and Neural Systems*. MIT Press, 1999.

[Mea89] C. Mead. *Pulsed Neural Networks*. VLSI systems series. Addison-Wesley, 1989.

[Mer14] P. A. Merolla. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197), August 2014.

[MF07] Y. Maeda and Y. Fukuda. FPGA implementation of pulse density Hopfield neural network. In *Neural Networks, 2007 (IJCNN 2007), International Joint Conference on*, pages 700–704, August 2007.

[MH03] M. Milev and M. Hristov. Analog implementation of ANN with inherent quadratic nonlinearity of the synapses. *Neural Networks, IEEE Transactions on*, 14(5), September 2003.

[MM14] D. Maliuk and Y. Makris. An experimentation platform for on-chip integration of analog neural networks : A pathway to trusted and robust analog/RF ICs. *Neural Networks and Learning Systems, IEEE Transactions on*, PP(99) :1–1, 2014.

[NAY62] Jinichi Nagumo, Suguru Arimoto, and Shuji Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10) :2061–2070, 1962.

[Pal13] Günther Palm. Neural associative memories and sparse coding. *Neural Networks*, 37 :165–171, 2013.

[PDW89] M.J.M. Pelgrom, Aad C.J. Duinmaijer, and A.P.G. Welbers. Matching properties of MOS transistors. *Solid-State Circuits, IEEE Journal of*, 24(5) :1433–1439, Oct 1989.

[PPG+13] E. Painkras, L.A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D.R. Lester, A.D. Brown, and S.B. Furber. Spinnaker : A 1-w 18-core system-on-chip for massively-parallel neural network simulation. *Solid-State Circuits, IEEE Journal of*, 48(8) :1943–1953, Aug 2013.

[RH14] S. Ramakrishnan and J. Hasler. Vector-matrix multiply and winner-take-all as an analog classifier. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 22(2) :353–361, Feb 2014.

[Ros57] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[Ros61] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, DTIC Document, 1961.

[SBT+11] S. Saïghi, Y. Bornat, J. Tomas, G. Le Masson, and S. Renaud. A library of analog operators based on the Hodgkin-Huxley formalism for the design of tunable, real-time, silicon neurons. *Biomedical Circuits and Systems, IEEE Transactions on*, 5(1) :3–19, Feb 2011.

[UN95] K. Urahama and T. Nagao. K-winners-take-all circuit with o(n) complexity. *Neural Networks, IEEE Transactions on*, 6(3) :776–778, May 1995.

[WBLH69] David J Willshaw, O Peter Buneman, and Hugh Christopher Longuet-Higgins. Non-holographic associative memory. *Nature*, 1969.

[WD08] J. Wijekoon and P. Dudek. Compact silicon neuron circuit with spiking and bursting behaviour. *Elsevier Neural Networks*, 21 :524–534, Mar 2008.

[YC10] T. Yu and G. Cauwenberghs. Analog VLSI biophysical neurons and synapses with programmable membrane channel kinetics. *Biomedical Circuits and Systems, IEEE Transactions on*, 4(3) :139–148, June 2010.

Résumé

Les réseaux de neurones artificiels permettent de résoudre des problèmes que des processeurs classiques ne peuvent pas résoudre sans utiliser une quantité considérable de ressources matérielles. L'analyse et la classification de multiples signaux en sont des exemples. Ces réseaux sont de plus en plus implantés sur des circuits intégrés. Ils ont ainsi pour but d'augmenter les capacités de calcul de processeurs ou d'effectuer leur traitement dans des systèmes embarqués.

Dans un contexte d'application embarquée, la surface et la consommation d'énergie du circuit sont prépondérantes. Cependant, le nombre de connexions entre les neurones est élevé. De plus, les poids synaptiques ainsi que les fonctions d'activation utilisées rendent les implantations sur circuit complexes. Ces aspects, communs dans la plupart des modèles de réseaux de neurones, limitent l'intégration d'un réseau contenant un nombre de neurones de l'ordre de la centaine.

Le modèle des réseaux de neurones à cliques permet de réduire la densité de connexions au sein d'un réseau, tout en gardant une capacité de stockage d'information plus grande que les réseaux de Hopfield, qui est un modèle standard de réseaux de neurones. Ce modèle est donc approprié pour implanter un réseau de grande taille, à condition de l'intégrer de façon à garder la faible complexité de ses fonctions, pour consommer un minimum d'énergie.

Dans ce document, nous proposons un circuit mixte analogique/numérique implantant le modèle des réseaux de neurones à cliques. Nous proposons également plusieurs architectures de réseau pouvant contenir un nombre indéterminé de neurones. Cela nous permet de construire des réseaux de neurones à cliques contenant jusqu'à plusieurs milliers de neurones et consommant peu d'énergie.

Pour valider les concepts décrits dans ce document, nous avons fabriqué et testé un prototype d'un réseau de neurones à cliques contenant trente neurones sur puce. Nous utilisons pour cela la technologie Si CMOS 65 nm, avec une tension d'alimentation de 1 V. Le circuit a des performances de récupération de l'information similaires à celles du modèle théorique, et effectue la récupération d'un message en 58 ns. Le réseau de neurones occupe une surface de silicium de 16 470 µm² et consomme 145 µW. Ces mesures attestent une consommation d'énergie par neurone de 423 fJ au maximum. Ces résultats montrent que le circuit produit est dix fois plus efficace qu'un équivalent numérique en termes de surface de silicium occupée et de latence.

Mots-clés : Réseaux de neurones artificiels, Réseaux de neurones à cliques, Implantation mixte analogique/numérique

Abstract

Artificial neural networks solve problems that classical processors cannot solve without using a huge amount of resources. For instance, multiple-signal analysis and classification are such problems. Moreover, artificial neural networks are more and more integrated on-chip. They aim therefore at increasing processors computational abilities or processing data in embedded systems.

In embedded systems, circuit area and energy consumption are critical parameters. However, the amount of connections between neurons is very high. Besides, circuit integration is difficult due to weighted connections and complex activation functions. These limitations exist for most artificial neural networks models and are thus an issue for the integration of a neural network composed of a high number of neurons (hundreds of them or more).

Clique-based neural networks are a model of artificial neural networks reducing the network density, in terms of connections between neurons. Its information storage capacity is moreover greater than that of a standard artificial neural networks model such as Hopfield neural networks. This model is therefore suited to implement a high number of neurons on chip, leading to low-complexity and low-energy consumption circuits.

In this document, we introduce a mixed-signal circuit implementing clique-based neural networks. We also show several generic network architectures implementing a network of any number of neurons. We can therefore implement clique-based neural networks of up to thousands of neurons consuming little energy.
In order to validate the proposed implementation, we have fabricated a 30-neuron clique-based neural network prototype integrated on chip for the Si 65-nm CMOS 1-V supply process. The circuit shows decoding performances similar to the theoretical model and executes the message recovery process in 58 ns. Moreover, the entire network occupies a silicon area of 16,470 µm² and consumes 145 µW, yielding a measured energy consumption per neuron of 423 fJ maximum. These results show that the fabricated circuit is ten times more efficient in terms of occupied silicon area and latency than a digital equivalent circuit.

Keywords : Artificial neural networks, Clique-based neural networks, Mixed analog/digital circuit implementation

TELECOM Bretagne

n° d'ordre : 2015telb0367
Télécom Bretagne
Technopôle Brest-Iroise - CS 83818 - 29238 Brest Cedex 3
Tél : + 33(0) 29 00 11 11 - Fax : + 33(0) 29 00 10 00